

日本国特許庁
PATENT OFFICE
JAPANESE GOVERNMENT



別紙添付の書類に記載されている事項は下記の出願書類に記載されて
いる事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed
with this Office.

出願年月日
Date of Application:

1999年 1月14日

出願番号
Application Number:

平成11年特許願第007384号

出願人
Applicant(s):

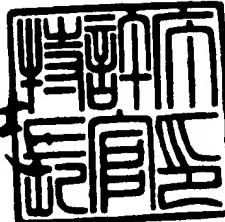
インターナショナル・ビジネス・マシーンズ・コーポレイシ
ョン

CERTIFIED COPY OF
PRIORITY DOCUMENT

1999年 4月30日

特許庁長官
Commissioner,
Patent Office

佐山 建



出証番号 出証特平11-3027520

【書類名】 特許願

【整理番号】 JA998232

【あて先】 特許庁長官 殿

【国際特許分類】 H03M 13/00

【発明者】

【住所又は居所】 神奈川県大和市下鶴間1623番地14 日本アイ・ビ
ー・エム株式会社 東京基礎研究所内

【氏名】 田村 哲也

【特許出願人】

【識別番号】 390009531

【住所又は居所】 アメリカ合衆国10504、ニューヨーク州アーモンク
(番地なし)

【氏名又は名称】 インターナショナル・ビジネス・マシーンズ・コーポレ
イション

【代理人】

【識別番号】 100086243

【弁理士】

【氏名又は名称】 坂口 博

【連絡先】 0462-73-3318、3325、3455

【選任した代理人】

【識別番号】 100091568

【弁理士】

【氏名又は名称】 市位 嘉宏

【手数料の表示】

【予納台帳番号】 024154

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9304391

【包括委任状番号】 9304392

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 標数2のガロア体上で定義される超楕円曲線のヤコビ多様体の群演算を実施する装置及び方法

【特許請求の範囲】

【請求項1】

$G F (2^n)$ 上で定義される超楕円曲線 $y^2 + y = f(x)$ のヤコビ多様体の因子 $D_1 = g.c.d. (a_1(x), y - b_1(x))$ 及び $D_2 = g.c.d. (a_2(x), y - b_2(x))$ に対し群演算を実施する装置であって、

$a_1(x), a_2(x), b_1(x)$ 及び $b_2(x)$ を格納する手段と、

$GCD (a_1(x), a_2(x)) = 1$ (GCD は最大公約多項式) である場合における $s_1(x) a_1(x) + s_2(x) a_2(x) = 1$ となる $s_1(x)$ 又は $s_2(x)$ を用いて、 $q(x) = \{ s_1(x) (b_1(x) + b_2(x)) \} \bmod a_2(x)$ 又は $q(x) = \{ s_2(x) (b_1(x) + b_2(x)) \} \bmod a_1(x)$ を計算する手段と、

を有することを特徴とする装置。

【請求項2】

$G F (2^n)$ 上で定義される超楕円曲線 $y^2 + y = f(x)$ のヤコビ多様体の因子 $D_1 = g.c.d. (a_1(x), y - b_1(x))$ 及び $D_2 = g.c.d. (a_2(x), y - b_2(x))$ に対し、 $D_1 + D_2$ と線形同値なりデュースト・ディバイザ $D' = g.c.d. (a'(x), y - b'(x))$ の $a'(x)$ 及び $b'(x)$ を計算する装置であって、

$GCD (a_1(x), a_2(x)) = 1$ (GCD は最大公約多項式) である場合における $s_1(x) a_1(x) + s_2(x) a_2(x) = 1$ となる $s_1(x)$ を用いて、 $q(x) = s_1(x) (b_1(x) + b_2(x)) \bmod a_2(x)$ を計算する手段と、

モニック化された $\alpha(x) = Q(q^2(x) a_1(x), a_2(x)) + Q(f(x), a_1(x) a_2(x))$ ($Q(A, B)$ は A を B で割ったときの商) を計算する手段と、

$\beta(x) = (q(x) a_1(x) + b_1(x) + 1) \bmod \alpha(x)$ を計算する手段と、

$a'(x) = Q(f(x) + \beta^2(x), \alpha(x))$ を計算する手段と、

$b'(x) = (\beta(x) + 1) \bmod a'(x)$ を計算する手段と、

を有する装置。

【請求項3】

$G F (2^n)$ 上で定義される超橿円曲線 $y^2 + y = f(x)$ のヤコビ多様体の因子 $D_1 = g.c.d. (a_1(x), y - b_1(x))$ に対し群演算を実施する装置であって、 $a_1(x)$ 及び $b_1(x)$ を格納する手段と、
 $q(x) = Q(b_1^2(x) + f(x) \bmod a_1^2(x), a_1(x))$ ($Q(A, B)$ は A を B で割ったときの商) を計算する手段、
 を有することを特徴とする装置。

【請求項4】

$G F (2^n)$ 上で定義される超橿円曲線 $y^2 + y = f(x)$ のヤコビ多様体の因子 $D_1 = g.c.d. (a_1(x), y - b_1(x))$ に対し、 $D_1 + D_1$ と線形同値なりデュースト・ディバイザ $D' = g.c.d. (a'(x), y - b'(x))$ の $a'(x)$ 及び $b'(x)$ を計算する装置であって、

$q(x) = Q(b_1^2(x) + f(x) \bmod a_1^2(x), a_1(x))$ ($Q(A, B)$ は A を B で割ったときの商) を計算する手段と、

モニック化された $\alpha(x) = q^2(x) + Q(f(x), a_1^2(x))$ を計算する手段と、

$\beta(x) = (b_1^2(x) + f(x) \bmod a_1^2(x) + 1) \bmod \alpha(x)$ を計算する手段と、

$a'(x) = Q(f(x) + \beta^2(x), \alpha(x))$ を計算する手段と、

$b'(x) = (\beta(x) + 1) \bmod a'(x)$ を計算する手段と、

を有する装置。

【請求項5】

$G F (2^n)$ 上で定義される超橿円曲線 $y^2 + y = f(x)$ のヤコビ多様体の因子 $D_1 = g.c.d. (a_1(x), y - b_1(x))$ 及び $D_2 = g.c.d. (a_2(x), y - b_2(x))$ に対し、 $D_1 + D_2$ と線形同値なりデュースト・ディバイザ $D' = g.c.d. (a'(x), y - b'(x))$ の $a'(x)$ 及び $b'(x)$ を計算する方法であって、

$GCD(a_1(x), a_2(x)) = 1$ (GCD は最大公約多項式) である場合における $s_1(x) a_1(x) + s_2(x) a_2(x) = 1$ となる $s_1(x)$ を用いて、 $q(x) = \{ s_1(x) (b_1(x) + b_2(x)) \} \bmod a_2(x)$ を計算し且つ記憶装置に格納するステップと、

モニック化された $\alpha(x) = Q(q^2(x) a_1(x), a_2(x)) + Q(f(x), a$

$a_1(x) a_2(x)$) ($Q(A, B)$ は A を B で割ったときの商) を計算し且つ記憶装置に格納するステップと、

$\beta(x) = (q(x) a_1(x) + b_1(x) + 1) \bmod \alpha(x)$ を計算し且つ記憶装置に格納するステップと、

$a'(x) = Q(f(x) + \beta^2(x), \alpha(x))$ を計算し且つ記憶装置に格納するステップと、

$b'(x) = (\beta(x) + 1) \bmod a'(x)$ を計算し且つ記憶装置に格納するステップと、

を含む方法。

【請求項6】

$GF(2^n)$ 上で定義される超精円曲線 $y^2 + y = f(x)$ のヤコビ多様体の因子 $D_1 = g.c.d. (a_1(x), y - b_1(x))$ に対し、 $D_1 + D_1$ と線形同値なリデュースト・ディバイザ $D' = g.c.d. (a'(x), y - b'(x))$ の $a'(x)$ 及び $b'(x)$ を計算する方法であって、

$q(x) = Q(b_1^2(x) + f(x) \bmod a_1^2(x), a_1)$ ($Q(A, B)$ は A を B で割ったときの商) を計算し且つ記憶装置に格納するステップと、

モニック化された $\alpha(x) = q^2(x) + Q(f(x), a_1^2(x))$ を計算し且つ記憶装置に格納するステップと、

$\beta(x) = (b_1^2(x) + f(x) \bmod a_1^2(x) + 1) \bmod \alpha(x)$ を計算し且つ記憶装置に格納するステップと、

$a'(x) = Q(f(x) + \beta^2(x), \alpha(x))$ を計算し且つ記憶装置に格納するステップと、

$b'(x) = (\beta(x) + 1) \bmod a'(x)$ を計算し且つ記憶装置に格納するステップと、

を含む方法。

【請求項7】

$GF(2^n)$ 上で定義される超精円曲線 $y^2 + y = f(x)$ のヤコビ多様体の因子 $D_1 = g.c.d. (a_1(x), y - b_1(x))$ 及び $D_2 = g.c.d. (a_2(x), y - b_2(x))$ に対し群演算を実施する方法であって、

$a_1(x)$ 、 $a_2(x)$ 、 $b_1(x)$ 及び $b_2(x)$ を格納するステップと、
 $\text{GCD}(a_1(x), a_2(x)) = 1$ である場合における $s_1(x)a_1(x) + s_2(x)a_2(x) = 1$ となる $s_1(x)$ 又は $s_2(x)$ を用いて、 $q(x) = s_1(x)(b_1(x) + b_2(x)) \bmod a_2(x)$ 又は $q(x) = \{s_2(x)(b_1(x) + b_2(x))\} \bmod a_1(x)$ を計算し且つ記憶装置に格納するステップと、

を含む、方法。

【請求項 8】

$GF(2^n)$ 上で定義される超楕円曲線 $y^2 + y = f(x)$ のヤコビ多様体の因子 $D_1 = \text{g.c.d.}(a_1(x), y - b_1(x))$ に対し群演算を実施する方法であって、

$a_1(x)$ 及び $b_1(x)$ を格納するステップと、

$q(x) = Q(b_1^2(x) + f(x) \bmod a_1^2(x), a_1)$ ($Q(A, B)$ は A を B で割ったときの商) を計算し且つ記憶装置に格納するステップ、

を含む、方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、 $GF(2^n)$ 上で定義される超楕円曲線 $y^2 + y = f(x)$ のヤコビ多様体の因子 $D_1 = \text{g.c.d.}(a_1(x), y - b_1(x))$ 及び $D_2 = \text{g.c.d.}(a_2(x), y - b_2(x))$ に対し群演算を実施する装置及び方法に関する。

【0002】

【従来の技術】

本願は、 $GF(2^n)$ 上で定義される超楕円曲線のヤコビ多様体の群演算をハードウェアで実行するのに適したアルゴリズムを開示する。以下、本発明を理解する上で必要な前提知識について説明する。

【0003】

[1] 超楕円曲線及び因子

体を K 、その代数的閉体を $K\bar{}$ （ K の上にバーが付いたもの）とする。 K 上、種数 g の超楕円曲線 C は式： $y^2 + h(x)y = f(x)$ で定義される。ここで $h(x)$ は、次数 g 以下の多項式であり、 $f(x)$ は $2g + 1$ 以下のモニック多項式である。

多項式 f 及び h の係数は K の元である。曲線 C は特異点を持たないものとする。また有理点 $P = (x, y)$ が与えられた時、 P の共役点を $P^- = (x, -y - h(x))$ と定義する (P^- は P の上にバーが付いたもの)。もし P が無限遠点 P_∞ ならば、 $P_\infty = P_\infty^-$ (P_∞^- は P_∞ の上にバーが付いたもの) とする。以後、本願では体 $K = GF(2^n)$ 、 $h(x) = 1$ の場合について考える。

【0004】

曲線 C 上の有限個の K 有理点 P_1, \dots, P_r についての整数係数の形式的な和 $m_1 P_1 + m_2 P_2 + \dots + m_r P_r$ を考えて、これを C の因子 (divisor) と呼ぶ。 C の因子は一般的には

【数1】

$$D = \sum_{P_i \in C} m_i P_i$$

で与えられる。因子 D の次数は、 $\deg D = \sum m_i$ とする。 C の因子

【数2】

$$D_1 = \sum_{\substack{P_i \in C \\ P_i \in C}} m_i P_i$$

【数3】

$$D_2 = \sum_{\substack{P_i \in C \\ P_i \in C}} n_i P_i$$

に対して、

【数4】

$$D_1 + D_2 = \sum_{\substack{P_i \in C \\ P_i \in C}} (m_i + n_i) P_i$$

と定義することにより、 C の因子全体の集合 D (C) は加群となり、これを因子群と呼ぶ。次数が 0 の因子全体は因子群の部分群となり、これを D^0 (C) と記

す。曲線Cの0でない有理関数hの零点及び極は高々有限個であることより、hの零点及び極を用いてhの因子

(h)

を

【数5】

$$\text{div}(h) = \sum_{P_i \in C} \text{ord}_{P_i}(h) P_i = \sum m_i P_i - \sum n_i Q_i$$

で定義する。ここで P_i は有理関数hの零点であり、 m_i はその重複度、 Q_i は有理関数hの極であり、 n_i は極の重複度、 $\text{ord}_{P_i}(h)$ は有理関数hの点 P_i における位数である。ある有理関数の因子となるような因子を主因子(principal divisor)という。主因子全体の集合を主因子群と呼び、 $D^1(C)$ と表す。

【0005】

一般に、有理関数の零点の個数と極の個数は重複度(位数)を込めて考えると等しいので $D^1(C) \subset D^0(C)$ である。2つの因子 D_1 (数1), D_2 (数2) $\in D^0(C)$ が与えられた時、2つの因子のg.c.d.(D_1, D_2)を、 $\sum \min(m_i, n_i) P_i - (\sum \min(m_i, n_i) P_\infty)$ で定義する。)。また式より明らかにg.c.d.(D_1, D_2) $\subset D^0(C)$ である。

【0006】

[2] ヤコビ多様体の定義

ヤコビ多様体は、Cの次数が0の因子全体のなす群 $D^0(C)$ の主因子群 $D^1(C)$ に関する剰余群で定義される(山本芳彦, “数論入門2” 岩波書店(1996)等を参照のこと)。これを $J(C)$ と表す。もし、 $D_1, D_2 \in D^0(C)$ で $D_1 - D_2 \in D^1(C)$ であるならば、 D_1, D_2 は線形同値 $D_1 \sim D_2$ であるという。 $\forall D \in D^0(C)$ は、次の条件を満たす因子 D_1

【数6】

$$D_1 = \sum_{P_i \in C} m_i P_i - (\sum_{P_i \in C} m_i) P_\infty$$

$(m_i \geq 0)$ に変形することができる。

(1) $D_1 \sim D$

(2) $P_i = (x_i, y_i)$ と共に P_i^- (P_i の上にバーが付いたもの) は同時に現れない。

(3) $P_i = P_i^-$ である時は、 $m_i = 1$

このように変形された因子をセミリデュースト・ディバイザ (semi-reduced divisor) と呼ぶ。さらに semi-reduced divisor に対し、

【数7】

$$\sum_{P_i \in C} m_i \leq g$$

の条件を与えることにより、ヤコビ多様体の元は一意に表現される。このように変形された因子をリデュースト・ディバイザ (reduced divisor) と呼ぶ。

【0007】

任意の semi-reduced divisor D は、 $D = g.c.d. (a(x), y - b(x))$ によって一意に表すことができる。ここで $a(x) = \prod_i (x - x_i)^{m_i}$ 、 $b(x)$ は $b(x_i) = y_i$ 、 $\deg b < \deg a$ を満たす唯一の多項式である。 D が reduce

d divisorである必要十分条件は、 $\deg a \leq g$ である。以降、D.G.Cantor, "Computing in the Jacobian of a Hyperelliptic Curve," Math. of Comp., 48, No.177, pp.95-101, (1987) にならって、g.c.d. ($a(x)$, $y - b(x)$) を $\text{div}(a, b)$ と表す。また、これ以後因子Dは2つの多項式a及びbと同一視する。

【0008】

$J(C; GF(2^n))$ 上の離散対数問題とは、 $D_1, D_2 \in J(C; GF(2^n))$ に対して、 $D_1 = m D_2$ となる整数mを求める問題である。

【0009】

[3] ヤコビ多様体の安全条件

安全な超楕円曲線暗号を構成するために、ヤコビ多様体 $J(C; GF(2^n))$ が満たさなければならない条件は、酒井康行、石塚裕一、櫻井幸一、"安全な超楕円曲線暗号の構成とその実装," SCIS'98-10.1.B, Jan. 1998等によると、次のとおりである。

C1 $\# J(C; GF(2^n))$ が大きな素数で割切れる。

C2 $J(C; GF(2^n))$ の最大素因数が $(2^n)^k - 1$ ($k < (\log 2^2)^2$) を割り切らない。

C3 $2g + 1 < \log 2^n$

【0010】

[4] ヤコビ多様体の群演算のアルゴリズム

ヤコビ多様体における加算は、 $D_1, D_2 \in J(C; GF(2^n))$ に対し、次に $D_1 + D_2$ と線形同値な reduced divisor D' を求めることである。Cantorの先の論文及び N.Koblitz, "Hyperelliptic curve cryptosystems," Journal of Cryptology, 1, pp.139-150, (1989)によれば、加算のアルゴリズムは2つの手続きから成る。この手続き1では、入力 $D_1 = \text{div}(a_1, b_1)$ 及び $D_2 = \text{div}(a_2, b_2)$ に対し、 $D_1 + D_2 \sim D$ ($D = \text{div}(a, b)$) となるような semi-reduced divisor D を求める。手続き2ではこの D を入力として $D \sim D'$ ($D' = \text{div}(a', b')$, $\deg b' < \deg a'$, $\deg a' \leq g$) となるような reduced divisor D' を求める。超楕円曲線を $y^2 + h(x)y = f(x)$ とすると、これらの

手続きは次のとおりになる。

【0011】

手続き 1

Input $a_1, b_1 D_1 = \text{div}(a_1, b_1)$

$a_2, b_2 D_2 = \text{div}(a_2, b_2)$

Output a, b

(1) $a_1(x), a_2(x), b_1(x) + b_2(x) + h(x)$ の最大公約多項式 (GCDと記する)
 $d = d(x)$ であって、 $d = s_1 a_1 + s_2 a_2 + s_3 (b_1 + b_2 + h)$ を満たす $s_1(x), s_2(x), s_3(x)$ を求める。

(2) $a(x), b(x)$ を次式に基づいて計算する。

$$a = a_1 a_2 / d^2$$

$$b = (s_1 a_1 b_2 + s_2 a_2 b_1 + s_3 (b_1 b_2 + f)) / d \bmod a$$

手続き 2

Input a, b

Output $a', b' D \sim D'$

(1) 次式に基づいて、 $a'(x)$ 及び $b'(x)$ を計算する。

$$a' = (f - h b - b^2) / a$$

$$b' = (-h - b) \bmod a'$$

(2) **if** (**deg** $a' > g$) **then**

$$a = a'$$

$$b = b'$$

goto (1)

else **end**

特に 2 倍算では手続き 1 は次のように簡単化できる。

手続き1

$$a = a_1^2$$

$$b = (b_1^2 + f) \bmod a$$

goto 手続き2 (1)

【0012】

【発明が解決しようとする課題】

上で述べたようなアルゴリズムにてそのまま計算を実施すると、次数の高い多項式の演算が必要となり、計算量が多くなるという欠点がある。

【0013】

よって、本発明の目的は、ヤコビ多様体における群演算を少ない演算量にて実現することとする。

【0014】

また、ヤコビ多様体における群演算を少ないハードウェア量にて実施可能とすることも、本発明の目的である。

【0015】

【課題を解決するための手段】

上でも引用した論文に記載されているようにKoblitzは、種数が1より大きい超楕円曲線のヤコビ多様体上での離散対数問題を用いた暗号を提案した。しかし、種数が2の曲線はFreyによって安全でないことが示された (G. Frey, H.G. Ruck, "A Remark Concerning m -Divisibility and the Discrete Logarithm in the Divisor Class Group of Curves," Math. of Comp., 62, No. 206, pp.865-874, (1994) を参照のこと)。種数が3以上の曲線では、安全と思われる曲線がいくつか見つかっている (酒井康行, 石塚裕一, 櫻井幸一, "安全な超楕円曲線暗号の構成とその実装," SCIS' 98-10.1.B, Jan. 1998、北村出, "小さい標数の有限体上でヤコビ多様体が almost prime となる超楕円曲線," SCIS' 98-7.1.A, Jan. 1998、S. Arita, "Public Key Cryptosystems with Cab curve(1)," IEICE ISEC97-54 pp. 13-23 (1997)等を参照のこと)。

一般に $GF(2^n)$ における演算は (1) 加算・乗算が比較的小規模なハードウェアで高速に実行できる、(2) 2乗演算が簡単にできる、(3) 逆元演算が伊東一辻井の提案した方法によって高速に実行できる (T. Itoh, S. Tsujii, "A Fast Algorithm for Computing Multiplicative Inverse in $GF(2^m)$ Using Normal Bases," *Inform. and Comput.*, vol.83, No. 1, pp.171-177, (1989)) などの理由でハードウェア実装に適する。さらに使用する定義体が楕円曲線暗号と比べて小さくて済むこと、上で述べた、Cantorのアルゴリズムにおいて行われる最大公約多項式を求める計算が複数個の乗算器を並列実行させることによって効率的に行うことことができるから、超楕円曲線暗号は楕円曲線暗号よりハードウェア実装にむいている。よって、本発明では、Cantorのアルゴリズムを改良して、計算量及びハードウェア量を減らす。

【0016】

そのため以下のような特徴がある。すなわち、 $GF(2^n)$ 上で定義される超楕円曲線 $y^2 + y = f(x)$ のヤコビ多様体の因子 $D_1 = g.c.d. (a_1(x), y - b_1(x))$ 及び $D_2 = g.c.d. (a_2(x), y - b_2(x))$ ($g.c.d.$ は段落番号 0005 に定義されている) に対し群演算を実施する装置は、 $a_1(x)$ 、 $a_2(x)$ 、 $b_1(x)$ 及び $b_2(x)$ を格納する手段と、 $GCD(a_1(x), a_2(x)) = 1$ (GCD は最大公約多項式) である場合における $s_1(x)a_1(x) + s_2(x)a_2(x) = 1$ となる $s_1(x)$ を用いて、 $q(x) = \{s_1(x)(b_1(x) + b_2(x))\} \bmod a_2(x)$ を計算する手段と有する。このように新たな関数 $q(x)$ を設けることにより、全体の計算量が減少し、且つハードウェア量も少なくて済む。実施例では、超楕円曲線が $y^2 + y = x^7$ である例が詳しく述べられているが、このような曲線でない場合であっても、この $q(x)$ は有効に用いることができる。なお、群演算では交換則が成立するので、 a_1 、 b_1 及び s_1 と、 a_2 、 b_2 及び s_2 とを交換して得られる $q(x)$ 等を用いても同じ解を得ることができる。これ以降煩雑を防ぐために片方のみを用いて説明することもあるが、交換しても同じ意味を有する。 $q(x) = \{s_1(x)(b_1(x) + b_2(x))\} \bmod a_2(x)$ は、 $q(x) = \{s_2(x)(b_1(x) + b_2(x))\} \bmod a_1(x)$ とすることができる。

【0017】

なお、 $D_1 = D_2$ の場合には、 $a_1(x)$ 及び $b_1(x)$ を格納する手段と、 $q(x) = Q(b_1^2(x) + f(x) \bmod a_1^2(x), a_1)$ ($Q(A, B)$ は A を B で割ったときの商) を計算する手段を設ける。このように別個の $q(x)$ を定義する。

【0018】

超楕円曲線が $y^2 + y = f(x)$ である場合に、 $GF(2^n)$ 上で定義される超楕円曲線のヤコビ多様体の因子 $D_1 = g.c.d. (a_1(x), y - b_1(x))$ 及び $D_2 = g.c.d. (a_2(x), y - b_2(x))$ に対し、 $D_1 + D_2$ と線形同値なリデュースト・ディバイザ $D' = g.c.d. (a'(x), y - b'(x))$ の $a'(x)$ 及び $b'(x)$ を計算する装置は、 $GCD(a_1(x), a_2(x)) = 1$ (GCD は最大公約多項式) である場合における $s_1(x) a_1(x) + s_2(x) a_2(x) = 1$ となる $s_1(x)$ を用いて、 $q(x) = s_1(x) (b_1(x) + b_2(x)) \bmod a_2(x)$ を計算する手段と、モニック化された $\alpha(x) = Q(q^2(x) a_1(x), a_2(x)) + Q(f(x), a_1(x) a_2(x))$ (又は、 $\alpha(x) = Q(q^2(x) a_2(x), a_1(x)) + Q(f(x), a_1(x) a_2(x))$) ($Q(A, B)$ は A を B で割ったときの商) を計算する手段と、 $\beta(x) = (q(x) a_1(x) + b_1(x) + 1) \bmod \alpha(x)$ (又は、 $\beta(x) = (q(x) a_2(x) + b_2(x) + 1) \bmod \alpha(x)$) を計算する手段と、 $a'(x) = Q(f(x) + \beta^2(x), \alpha(x))$ を計算する手段と、 $b'(x) = (\beta(x) + 1) \bmod a'(x)$ を計算する手段とを有する。

【0019】

一方、 $D_1 = D_2$ である場合には、 $q(x) = Q(b_1^2(x) + f(x) \bmod a_1^2(x), a_1)$ ($Q(A, B)$ は A を B で割ったときの商) を計算する手段と、モニック化された $\alpha(x) = q^2(x) + Q(f(x), a_1^2(x))$ を計算する手段と、 $\beta(x) = (b_1^2(x) + f(x) \bmod a_1^2(x) + 1) \bmod \alpha(x)$ を計算する手段と、 $a'(x) = Q(f(x) + \beta^2(x), \alpha(x))$ を計算する手段と、 $b'(x) = (\beta(x) + 1) \bmod a'(x)$ を計算する手段とを有する。

【0020】

なお、以上はハードウェア化を前提とした構成であるが、これらをコンピュータ・プログラム等で実行するように変形することも可能である。その際、プログラムは、フロッピー・ディスクや CD-ROM 等の記憶媒体及び他の記憶装置に

格納される。

【0021】

【発明の実施の形態】

まず、本発明の基本的なアルゴリズムを説明する。

【0022】

なお、加算を行う時には a_1 と a_2 の最大公約多項式を求めなければならない。しかし、定義体が大きく且つ a_1 と a_2 が任意に選ばれた座標から与えられる時には、多くの場合 $\text{GCD}(a_1, a_2) = 1$ となる。それゆえ a_1 と a_2 が素でない場合の処理は性能に大きな影響を与えないもので、本願では $\text{GCD}(a_1, a_2) \neq 1$ の場合を以後扱わない。また、その最大公約多項式 $\text{GCD}(a_1, a_2) = 1$ は多項式 s_1, s_2 によって、 $s_1(x)a_1(x) + s_2(x)a_2(x) = 1$ のように表されるものとする。なお、素でない場合の計算においても、後で説明する手続き 1 の簡単化に用いる補題 1 及び 2 を一般化したものと、手続き 2 の簡単化に用いられる関数 $Q(u, v)$ を使用することができる。

【0023】

又、上で説明した従来技術の手続き 1 (1) の演算、すなわち、多項式 s_1, s_2 と最大公約多項式を求めるのには通常ユークリッド法が用いられる。ユークリッド法は、リード・ソロモン (Reed-Solomon) 符号などの復号で、誤り位置多項式・誤り数値多項式を求める時に使用され、多くの実装が行われている。例えば、特開平7-202718号公報又は特開昭62-122332号公報を参照のこと。よって、本発明では、手続き 1 (1) の 2 つの多項式の最大公約多項式 d と $d = s_1a_1 + s_2a_2$ を満たす s_1 (又は s_2) のみを求める処理は、計算できたものとして扱う。また、例えば以下で述べる実装例 (図 1) を用いて計算することもできる。実際の動作の詳細を説明するのは省略するが、 $s_1(x)$ は $Y \text{ reg}$ に outputされるものとする。また最大公約多項式 d が 1 となるよう $s_1(x)$ は正規化されているものとする。

【0024】

a_1 と a_2 が素である場合について、ヤコビ多様体の群演算のアルゴリズム (従来技術) を手続き 1 (2) 以降について変形すると次のとおりとなる。

変形1 (通常の加算の場合)

Input a_1, a_2, b_1, b_2 Output a', b'

$$a_3(x) = a_1 a_2$$

$$b_3(x) = (s_1 a_1 b_2 + s_2 a_2 b_1) \bmod a_3$$

$$a_4(x) = (f + b_3 + b_3^2) \bmod a_3$$

a₄(x)をモニック化 (最高次の項の係数を1にする)

$$b_4(x) = (b_3 + 1) \bmod a_4(x)$$

while (deg a₄(x) > g) {

$$a' = a_5(x) = (f + b_4 + b_4(x)^2) \bmod a_4(x)$$

$$b' = b_5(x) = (b_4 + 1) \bmod a_5(x)$$

$$a_4(x) = a'$$

$$b_4(x) = b'$$

} end

このアルゴリズムに対し $f(x) = x^7$ を用いる。群演算のアルゴリズムの手続き2 (1) では a 及び b の次数は2下がり、それ以後1づつ次数が下がることから、while loopの中を1回実行することにより、次数が3以下の a' が求まることが分かる。 b_3 の計算で剩余演算される多項式は7次式、 $a_4(x)$ の計算で被除多項式は10次式となり、多くの計算が必要となる。これを減らすために新しい多項式 $q(x) = s_1(b_1 + b_2) \bmod a_2$ を導入する。

【0025】

補題1

変形1の最初の $a_4(x)$ は、 $q(x)$ を用いて $a_4(x) = Q(q^2 a_1, a_2)$ で与えられる。ここで $Q(u, v)$ は u/v の商を与える関数である。

(証明)

まず、 $b_3(x) = q a_1 + b_1$ であることを示す。上で述べた仮定より、 $s_1 a_1 + s_2 a_2 = 1$ 。これを用いて b_3 の計算を $\deg a_1 a_2 > \deg b_1$ に注意して書き直すと、

$$\begin{aligned}
 b_3(x) &= (s_1 a_1 b_2 + s_2 a_2 b_1) \bmod a_3 \\
 &= (s_1 a_1 b_2 + (1 + s_1 a_1) b_1) \bmod a_1 a_2 \\
 &= (s_1 a_1 (b_1 + b_2)) \bmod a_1 a_2 + b_1 \\
 &= \{ (s_1 (b_1 + b_2)) \bmod a_2 \} a_1 + b_1 \\
 &= q a_1 + b_1 \cdots \cdots \cdots (\#)
 \end{aligned}$$

次に a_4 の割り算は割り切ることと、 $\deg b_3 < \deg a_3$ より、 $Q(b_3, a_3) = 0$ であることから、

$$\begin{aligned}
 a_4 &= Q(f + b_3 + b_3^2, a_3) \\
 &= Q(f, a_3) + Q(b_3^2, a_3)
 \end{aligned}$$

である。第2項は (#) を代入し、 $\deg b_1^2 < \deg a_3$ より $Q(b_1^2, a_3) = 0$ であることに注意すると、

$$Q(b_3^2, a_3) = Q(q^2 a_1^2 + b_1^2, a_3) = Q(q^2 a_1^2, a_3)$$

これより、 $a_4(x) = Q(q^2 a_1, a_2) + Q(f, a_3)$ Q.E.D.

【0026】

入力多項式を、

$$a_1(x) = x^3 + c_2 x^2 + c_1 x + c_0$$

$$a_2(x) = x^3 + e_2 x^2 + e_1 x + e_0$$

$$b_1(x) = d_2 x^2 + d_1 x + d_0$$

$$b_2(x) = f_2 x^2 + f_1 x + f_0$$

と定義し、 $f(x) = x^7$ を用いると、 a_4 の第2項は、 $Q(x^7, a_3) = x + c_2 + e_2$ となる。これより、 $q(x)$ を用いて変形1を書き直すと、以下のような本発明のアルゴリズムとなる。

【0027】

本発明のアルゴリズム（加算）Input a_1, a_2, b_1, b_2 Output a', b'

$$q(x) = s_1 (b_1 + b_2) \bmod a_2$$

$$a_4(x) = Q (q^2 a_1, a_2) + x + c_2 + e_2$$

a₄をモニック化（最高次数の項の係数を1にする）

$$b_4(x) = (q a_1 + b_1 + 1) \bmod a_4$$

if (deg a₄ > 3) then

$$a' = a_5(x) = Q (x^7 + b_4^2, a_4)$$

$$b' = b_5(x) = (b_4 + 1) \bmod a_5$$

else $a' = a_4, b' = b_4$

end

【0028】

なお $a_5(x)$ の計算で、 $\deg b_4 < \deg a_4$ であることより $Q(b_4, a_4) = 0$ であることを用いている。このアルゴリズムでは、次数の高い $a_3(x)$ が消えて、その剩余演算及び除算を行う必要がなくなっている。また $a_4(x)$ を計算するのに必要な乗算は、 Q の中の $q^2 a_1$ の次数が7次であることと標数が2であることより、9回でよい。さらに $a_5(x)$ の計算において必要なない $b_4(x)$ が Q の中から落とされている。これらにより大幅な計算回数の削減が可能となる。

【0029】

次に2倍算について考える。前と同様に従来技術の手続き1(2)を変更すると、次の変形2を得る。

変形2Input a_1, b_1 Output a', b'

$$a_3(x) = a_1^2$$

$$b_3(x) = (b_1^2 + f) \bmod a_3$$

```

 $a_4(x) = (f + b_3 + b_3^2) / a_3$ 
 $a_4(x)$  のモニック化 (最高次数の項の係数を 1 にする)
 $b_4(x) = (b_3 + 1) \bmod a_4$ 
while ( $\deg a_4 > g$ ) {
   $a' = a_5(x) = (f + b_4 + b_4^2) / a_4$ 
   $b' = b_5(x) = (b_4 + 1) \bmod a_5$ 
   $a_4 = a', b_4 = b'$ 
}
end

```

【0030】

通常の加算の場合と同様に、 a_4 の計算で被除多項式は 10 次式となり、多くの計算が必要となる。これを減らすために、 $q(x) = Q(b_3, a_1)$ を導入する。

補題 2

変形 2 の $a_4(x)$ は $q(x)$ を用いて、 $a_4(x) = q^2 + Q(f, a_3)$ で与えられる。

(証明)

$\deg b_3 < \deg a_3$ より $Q(b_3, a_3) = 0$ であるから、
 $a_4(x) = Q(f, a_3) + Q(b_3^2, a_3)$
 である。第 2 項は $Q(b_3^2, a_3) = Q(b_3, a_3)^2$ となる。これは、 $b_3 = r_1 + s_1 / a_1$, $\deg s_1 < \deg a_1$ とすると標数 2 の体の上では、 $b_3 = r_1^2 + s_1^2 / a_1^2$, $\deg s_1^2 < \deg a_1^2$ が成り立つことからも明らかである。ゆえに、 $a_4(x) = q^2 + Q(f, a_3)$ Q. E. D.

【0031】

加算と同様に $f(x) = x^7$ を用いると、 $a_3(x) = a_1^2(x)$ の中に奇数次の項がないことから $Q(x^7, a_3) = x$ となる。これより $q(x)$ を用いて変形 2 を書き直すと次の本発明のアルゴリズム (2 倍算) を得る。

本発明のアルゴリズム（2倍算）Input a_1, b_1 Output a', b'

$$b_3(x) = b_1^2 + x (a_1 - x^3)^2$$

$$q(x) = Q(b_3, a_1)$$

$$a_4(x) = q^2 + Q(f, a_3)$$

a_4 をモニック化（最高次数の項の係数を1にする）

$$b_4(x) = (b_3 + 1) \bmod a_4$$

if ($\deg a_4 > g$) then

$$a' = a_5(x) = Q(x^7 + b_4^2, a_4)$$

$$b' = b_5(x) = (b_4 + 1) \bmod a_5$$

else $a' = a_4, b' = b_4$

end

【0032】

なお $b_3(x)$ の計算においては、 $x^7 \bmod a_3 = x (x^3)^2 \bmod a_3 = x (a_1 - (a_1 - x^3))^2 \bmod a_3 = x (a_1 - x^3)^2 \bmod a_3 = x (a_1 - x^3)^2$ 、 $\deg b_1^2 < \deg a_3$ であることから $b_1^2 \bmod a_3 = b_1^2$ となることを用いている。また、 $b_3(x)$ の計算結果をストアしておく必要はない。なぜなら 標数2のガロア体上で2乗器は小さなハードウェアで実現できるので、レジスタの代わりに2乗器を持つ方がサイズ的に有利だからである。特に正規基底を用いる場合にはビットシフトだけで実現できる。 $b_3(x)$ が必要な時は、 $a_1(x), b_1(x)$ を2乗器に入力し、その出力を直接用いればよい。

【0033】

通常の加算と同様に、このアルゴリズムでは、次数の高い $a_3(x)$ が消えて、その剩余演算及び除算を行う必要がなくなっている。また $a_4(x)$ の次数は4次で標数が2であることより、それを計算するのに2乗算のみ必要で乗算は必要ない。さらに $a_5(x)$ の計算においても、必要なない $b_4(x)$ が Q の中から落とされている。

【0034】

なお補題1及び2は $h(x) = 1$ 以外でも成り立つ。また、 $Q(f, a_3)$ は f の次数が $2g + 1$ 次、 a_3 の次数が $2g$ 次であることに注意すると容易に計算することができる。

【0035】

また、超楕円曲線は上で使った $y^2 + y = x^7$ 以外のものであってもよい。例えば、 $g = 3$ の場合には、 $K = GF(2^{61})$ $f(x) = x^7 + x + 1$ や、 $K = GF(2^{67})$ $f(x) = x^7 + 1$ 等がある。上のアルゴリズムで x^7 としていたところを、このような $f(x)$ に置き換えれば、新たに $q(x)$ を導入する効果はある。

【0036】

では、図1に上記アルゴリズムの実装例を示す。レジスタ群1はセレクタ1(3)とセレクタ2(9)に接続されている。セレクタ1(3)及びセレクタ2(9)は共に乗算器及び二乗器5とインバータ7に接続されている。セレクタ1(3)はレジスタ入力用のセレクタであり、セレクタ2(9)は乗算器及び二乗器及びインバータ入力用のセレクタである。なお、セレクタ1(3)とセレクタ2(9)と乗算器及び二乗器5とインバータ7は、コントローラ11によりそれらの動作が制御されている(図1の点線表示)。レジスタ群1は、ワークエリアとして且つ結果格納のために用いられるレジスタUreg、Xreg、Yreg及びZregと、入力となる $a_1(x)$ 、 $a_2(x)$ 、 $b_1(x)$ 及び $b_2(x)$ を各々格納するレジスタ a_1 、 a_2 、 b_1 及び b_2 を含む。なお、UregとXregの記憶位置は4つあり、残りのレジスタの記憶位置は3つである。また、図示していないが、加算器は乗算器及び二乗器5等に設けられており、コントローラ11により加算が指示されている場合には加算器を動作させる。

【0037】

では、本発明のアルゴリズム(通常の加算)を実施する際に、図1の回路がどのように動作するかを説明する。図2にレジスタ群1の初期状態を示す。前提として、 $s_1(x) = s_{12}x^2 + s_{11}x + s_{10}$ の各係数をYregが格納している。また、 a_1 、 a_2 、 b_1 及び b_2 は、それぞれ $a_1(x)$ 、 $a_2(x)$ 、 $b_1(x)$ 、 $b_2(x)$ の係数を格

納している。但し、最高次である3次の項の係数は1であるから、これらは格納する必要がない。すなわち、 a_1 は c_2 、 c_1 及び c_0 を、 a_2 は e_2 、 e_1 及び e_0 、 b_1 は d_2 、 d_1 及び d_0 、 b_2 は f_2 、 f_1 及び f_0 を格納している。

【0038】

まず $q(x) = s_1(b_1 + b_2) \bmod a_2$ を得るための計算を実施する。セレクタ2(9)は、以下の計算を実施すべく、レジスタ群1から必要な値を取り出し、乗算器及び二乗器5にそれらを入力する。

$$(1) \quad p_4 = (s_{12} \cdot b'_2) \quad [x^4 \text{の係数}]$$

$$p_3 = (s_{12} \cdot b'_1 + s_{11} \cdot b'_2) \quad [x^3 \text{の係数}]$$

$$p_2 = s_{12} \cdot b'_0 \quad [x^2 \text{の係数}]$$

但し、

$$\begin{aligned} & (b_1 + b_2) \\ &= (d_2 + f_2) x^2 + (d_1 + f_1) x + (d_0 + f_0) \\ &= b'_2 x^2 + b'_1 x + b'_0 \end{aligned}$$

としている。

【0039】

なお、

$$\begin{aligned} & s_1(b_1 + b_2) \\ &= (s_{12} \cdot b'_2) x^4 \\ &+ (s_{12} \cdot b'_1 + s_{11} \cdot b'_2) x^3 \\ &+ (s_{12} \cdot b'_0 + s_{11} \cdot b'_1 + s_{10} \cdot b'_2) x^2 \\ &+ (s_{11} \cdot b'_0 + s_{10} \cdot b'_1) x \\ &+ s_{10} \cdot b'_0 \end{aligned}$$

である。

よって(1)の計算は、 $s_1(b_1 + b_2)$ の4次及び3次の項の完全な係数と、2次の項の一部の係数の、計算である。これらの計算結果はセレクタ1(3)によりUregに格納される(図3:Uregのみ図示)。(1)のような計算をするのは、乗算器及び二乗器5の乗算器が一度に4つしか使えないという前提があるか

らで、Uregは4つの格納場所を有しているため、乗算器の数に制限が無ければ上位4つの項の係数を(1)で計算することも可能である。また、 a_2 の剩余計算を行うので、 $s_1 (b_1 + b_2)$ の、 a_2 の最高次の次数である3未満の次数の項はそのまま残ってしまう。よって、2次以下の項の係数を、剩余計算の後から加算するようにしても結果は同じである。

【0040】

次に、 $p_4 x^4 + p_3 x^3 + p_2 x^2$ に対して $a_2(x) = x^3 + e_2 x^2 + e_1 x + e_0$ の剩余計算及び $s_1 (b_1 + b_2)$ の0次の項の係数の計算 $s_{10} \cdot b'_0 = p_0$ を行う。よって、セレクタ2(9)は必要な値を取り出し、乗算器及び二乗器5にそれらを入力する。

$$(2) (p_4 x^4 + p_3 x^3 + p_2 x^2) \bmod a_2$$

$$s_{10} \cdot b'_0 [x^0 \text{の係数}]$$

【0041】

$(p_4 x^4 + p_3 x^3 + p_2 x^2) \bmod a_2$ をより詳しく記述すると以下のようになる。

$$p'_3 = (p_3 + p_4 \cdot e_2) [x^3 \text{の係数}]$$

$$p'_2 = (p_2 + p_4 \cdot e_1) [x^2 \text{の係数}]$$

$$p'_1 = p_4 \cdot e_0 [x \text{の係数}]$$

そして、

$$p'_0 = s_{10} \cdot b'_0 [x^0 \text{の係数}]$$

も実施する。

これらの計算結果をセレクタ1(3)はUregに格納する(図4:Uregのみ図示)。(2)のような計算を実施するのも乗算器及び二乗器5の乗算器の数が4つになっているからである。

【0042】

次に $p'_3 x^3 + p'_2 x^2 + p'_1 x + p'_0 \bmod a_2$ の計算を行う。これをさらに詳しく記述すると以下のようになるので、セレクタ2(9)は必要な値を取り出し

、乗算器及び二乗器5にそれらを入力する。

$$(3) p''_2 = (p'_2 + p'_3 \cdot e_2) [x^2 \text{の係数}]$$

$$p''_1 = (p'_1 + p'_3 \cdot e_1) [x^1 \text{の係数}]$$

$$p''_0 = (p'_0 + p'_3 \cdot e_0) [x^0 \text{の係数}]$$

これらの計算結果をセレクタ1(3)はUregに格納する(図5:Uregのみ図示)。

【0043】

以上の計算では $s_1 (b_1 + b_2)$ のうち、 $(s_{11} \cdot b'_1 + s_{10} \cdot b'_2) x^2 + (s_{11} \cdot b'_0 + s_{11} \cdot b'_1) x$ について考慮されていない。よって、以下のような計算をすべく、セレクタ2(9)は必要な値を取り出し、乗算器及び二乗器5にそれらを入力する。

$$(4) p''_2 + (s_{11} \cdot b'_1 + s_{10} \cdot b'_2) [x^2 \text{の係数}]$$

$$p''_1 + (s_{11} \cdot b'_0 + s_{11} \cdot b'_1) [x \text{の係数}]$$

これらの計算にて $q(x) = q_2 x^2 + q_1 x + q_0$ は求まった。セレクタ1(3)は、これらの計算結果をZregに格納する(図6:Zregのみ図示)。

【0044】

次に $a_4(x) = Q(q^2 a_1, a_2) + x + c_2 + e_2$ の計算を実施する。そのため、最初に $q^2 a_1$ の計算を行う。但し、 a_2 の商の計算であるから2次以下の項は計算する必要が無い。以下の計算を行うため、セレクタ2(9)は、必要な値をレジスタ群1から取り出し、乗算器及び二乗器5にそれらを入力する。

$$(1) p_7 = q_2^2 [x^7 \text{の係数}]$$

$$p_6 = q_2^2 \cdot c_2 [x^6 \text{の係数}]$$

$$p_5 = q_2^2 \cdot c_1 + q_1^2 [x^5 \text{の係数}]$$

$$p_4 = q_1^2 \cdot c_2 + q_2^2 \cdot c_0 [x^4 \text{の係数}]$$

ここでは $q^2 a_1$ の3次の項の計算は乗算器の数及びUregの記憶位置が足りないので実施しない。これらの計算結果をセレクタ1(3)は、Uregに格納する

(図7: Uregのみ図示)。

【0045】

なお、 $q^2 a_1$ は以下のようになっている。

$$\begin{aligned}
 q^2 a_1 = & \\
 q_2^2 x^7 + & \\
 q_2^2 \cdot c_2 x^6 + & \\
 (q_2^2 \cdot c_1 + q_1^2) x^5 + & \\
 (q_1^2 \cdot c_2 + q_2^2 \cdot c_0) x^4 + & \\
 (q_1^2 \cdot c_1 + q_0^2) x^3 + & \\
 (q_1^2 \cdot c_0 + q_0^2 \cdot c_2) x^2 + & \\
 q_0^2 c_1 x + q_0^2 c_0
 \end{aligned}$$

【0046】

(1) の計算の開始と共に、 q_2^2 の逆数の計算を開始する。このため、セレクタ2(9)は、乗算器及び二乗器5で計算された q_2^2 の結果をインバータ7に入力する。 $q^- = 1 / q_2^2$ とする。

【0047】

次に、 a_2 による商を求める計算を行う。これは a_2 による剩余計算を行っていくことにより行われる。よって、まず以下の計算を行うため、セレクタ2(9)は必要な値をレジスタ群1から取り出し、乗算器及び二乗器5に入力する。

$$\begin{aligned}
 (2) \quad p'_6 &= p_6 + p_7 \cdot e_2 \quad [x^6 \text{の係数}] \\
 p'_5 &= p_5 + p_7 \cdot e_1 \quad [x^5 \text{の係数}] \\
 p'_4 &= p_4 + p_7 \cdot e_0 \quad [x^4 \text{の係数}] \\
 p'_3 &= (q_1^2 \cdot c_1 + q_0^2) \quad [x^3 \text{の係数}]
 \end{aligned}$$

a_2 による商を計算する場合、 $p_7 x^4$ が最初に求まる項であるが、 p_7 は既に求まっており且つ最終的に求めようとしている a_4 はモニック化されるので、Uregに格納しておく必要はない。これらの計算結果をセレクタ1(3)は、Uregに

格納する（図8：Uregのみ図示）。

【0048】

さらに、 a_2 による剰余計算を行う。但し、この計算においては a_4 の3次の項の係数（モニック化前）も求まるので、合わせてUregに格納することになる。よって、以下のような計算を行うため、セレクタ2（9）は必要な値をレジスタ群1から取り出し、乗算器及び二乗器5に入力する。

$$(3) \quad p''_5 = p'_5 + p'_6 \cdot e_2 \quad [x^5 \text{の係数}]$$

$$p''_4 = p'_4 + p'_6 \cdot e_1 \quad [x^4 \text{の係数}]$$

$$p''_3 = p'_3 + p'_6 \cdot e_0 \quad [x^3 \text{の係数}]$$

なお、モニック化前の a_4 を以下のように記載する。

$$a_4(x) = a'_{44}x^4 + a'_{43}x^3 + a'_{42}x^2 + a'_{41}x + a'_{40}$$

但し、 $a'_{43} = p'_6$ [a_4 の3次の項の係数] となる。

セレクタ1（3）は、 p'_6 をUregから取り出し、これらの計算結果と共にUregに格納する（図9：Uregのみ図示）。

【0049】

さらに、 a_2 による剰余計算を行う。但し、この計算においては a_4 の2次の項の係数（モニック化前）も求まるので、合わせてUregに格納することになる。よって、以下のような計算を行うため、セレクタ2（9）は必要な値をレジスタ群1から取り出し、乗算器及び二乗器5に入力する。

$$(4) \quad p_{34} = p''_4 + p''_5 \cdot e_2 \quad [x^4 \text{の係数}]$$

$$p_{33} = p''_3 + p''_5 \cdot e_1 \quad [x^3 \text{の係数}]$$

但し、 $a'_{42} = p''_5$ [a_4 の2次の項の係数] となる。

セレクタ1（3）は、 p''_5 及び a'_{43} をUregから取り出し、これらの計算結果と共にUregに格納する（図10：Uregのみ図示）。

【0050】

さらに、 a_2 による剰余計算を行う。但し、この計算においては a_4 の1次の項の係数（モニック化前）も求まるので、合わせてUregに格納することになる。

また、 a_4 のQ以外の項の加算も行う。よって、以下のような計算を行うため、セレクタ2(9)は必要な値をレジスタ群1から取り出し、乗算器及び二乗器5に入力する。

$$(5) \quad p_{43} = p_{33} + p_{34} \cdot e_2 \quad [x^3 \text{の係数}]$$

$$a'_{41} = p_{34} + 1 \quad [a_4 \text{の1次の項の係数}]$$

セレクタ1(3)は、 a'_{42} 及び a'_{43} をUregから取り出し、これらの計算結果と共にUregに格納する(図11:Uregのみ図示)。

【0051】

次に、 a_4 の0次の項の係数(モニック化前)を計算するため及び a_4 のQ以外の項の加算も行うため、セレクタ2(9)は必要な値をレジスタ群1から取り出し、乗算器及び二乗器5に入力する。

$$(6) \quad a'_{40} = p_{43} + c_2 + e_2 \quad [a_4 \text{の定数項}]$$

セレクタ1(3)は、 a'_{42} 、 a'_{43} 及び a'_{41} をUregから取り出し、これらの計算結果と共にUregに格納する(図12:Uregのみ図示)。これにて a_4 のモニック化前の値が求まった。

【0052】

次に、 a_4 のモニック化を行う。 a_4 は4次であり、その係数は q_2^2 である。よって、段落番号0046で説明した計算の終了を待って、 q^- をUregの各係数に掛ける。すなわち、以下の計算を行うため、セレクタ2(9)はインバータ7とレジスタ群1から必要な値を取り出し、乗算器及び二乗器5に入力する。

$$(7) \quad a'_{43} \cdot q^-$$

$$a'_{42} \cdot q^-$$

$$a'_{41} \cdot q^-$$

$$a'_{40} \cdot q^-$$

セレクタ1(3)は、これらの計算結果をXregに格納する(図13:Xregのみ図示)。これにてモニック化された a_4 が求まった。

【0053】

次に $b_4(x) = (q \cdot a_1 + b_1 + 1) \bmod a_4$ を計算する。まず、 $(q \cdot a_1 + b_1 + 1)$ の計算を行うが、Uregの記憶位置の数及び乗算器の数の制限のため、以下のように計算する。なお、 a_4 は4次の式であるから、 $(q \cdot a_1 + b_1 + 1)$ の3次以下の項は、剩余計算の後に加算しても結果は同じである。セレクタ2(9)は必要な値をレジスタ群1から取り出して、乗算器及び二乗器5に入力する。

$$(8) \quad p_5 = q_2 [x^5 \text{の係数}]$$

$$p_4 = (q_2 \cdot c_2 + q_1) [x^4 \text{の係数}]$$

$$p_3 = (q_2 \cdot c_1 + q_1 \cdot c_2 + q_0) [x^3 \text{の係数}]$$

$$p_2 = (d_1 + q_1 \cdot c_0) [x^2 \text{の係数}]$$

セレクタ1(3)は、これらの計算結果をUregに格納する（図14：Uregのみ図示）。

【0054】

なお、

$$q \cdot a_1 + b_1 + 1 =$$

$$q_2 x^5 +$$

$$(q_2 c_2 + q_1) x^4 +$$

$$(q_2 c_1 + q_1 c_2 + q_0) x^3 +$$

$$(d_2 + q_2 c_0 + q_1 c_1 + q_0 c_2) x^2 +$$

$$(d_1 + q_1 c_0 + q_0 c_1) x +$$

$$d_0 + q_0 c_0 + 1$$

【0055】

そして、 a_4 による剩余を計算する。なお、この剩余計算により x^1 の項が出現するので、 $d_1 x$ の加算も行う。詳細に記述すると以下のようないくつかの計算を実行する。このため、セレクタ2(9)は必要な値をレジスタ群1から取り出し、乗算器及び二乗器5に入力する。

$$(9) \quad p'_4 = p_4 + p_5 \cdot a_{43} [x^4 \text{の係数}]$$

$$\begin{aligned}
 p'{}_3 &= p{}_3 + p{}_5 \cdot a{}_{42} [x^3 \text{の係数}] \\
 p'{}_2 &= p{}_2 + p{}_5 \cdot a{}_{41} [x^2 \text{の係数}] \\
 p'{}_1 &= p{}_1 + p{}_5 \cdot a{}_{40} + d{}_1 [x^1 \text{の係数}]
 \end{aligned}$$

セレクタ1 (3) は、これらの計算結果をUregに格納する（図15：Uregのみ図示）。

【0056】

再度 a_4 による剩余計算を行う。なお、この剩余計算により x^0 の項の係数が計算されるので、 $d_0 + 1$ の加算も行う。詳細に記述すると以下のようない算を実行する。このため、セレクタ2 (9) は必要な値をレジスタ群1から取り出し、乗算器及び二乗器5に入力する。

$$\begin{aligned}
 (10) \quad p''{}_3 &= p'{}_3 + p'{}_4 \cdot a{}_{43} [x^3 \text{の係数}] \\
 p''{}_2 &= p'{}_2 + p'{}_4 \cdot a{}_{42} [x^2 \text{の係数}] \\
 p''{}_1 &= p'{}_1 + p'{}_4 \cdot a{}_{41} [x^1 \text{の係数}] \\
 p''{}_0 &= p'{}_4 \cdot a{}_{40} + d{}_0 + 1 [x^0 \text{の係数}]
 \end{aligned}$$

セレクタ1 (3) は、これらの計算結果をUregに格納する（図16：Uregのみ図示）。

【0057】

次に $(q \cdot a_1 + b_1 + 1)$ のうち a_4 の剩余計算に影響がなく且つ (8) 乃至 (10) でまだ加算されていない項を加算する。より詳細に記述すると、以下のようない算を実施する。セレクタ2 (9) は必要な値をレジスタ群1から取り出し、乗算器及び二乗器5に入力する。

$$\begin{aligned}
 (11) \quad p{}_{32} &= p''{}_2 + c{}_1 \cdot q{}_1 [x^2 \text{の係数}] \\
 p{}_{31} &= p''{}_1 + c{}_0 \cdot q{}_1 [x^1 \text{の係数}]
 \end{aligned}$$

セレクタ1 (3) は、これらの計算結果をUregに格納する（図17：Uregのみ図示）。

【0058】

($q \cdot a_1 + b_1 + 1$) のうち a_4 の剩余計算に影響がなく且つ (8) 乃至 (11) でまだ加算されていない項を加算する。より詳細に記述すると、以下のような計算を実施する。セレクタ2 (9) は必要な値をレジスタ群1から取り出し、乗算器及び二乗器5に入力する。

$$(12) \quad b_{42} = p_{32} + c_2 \cdot q_0 \quad [x^2 \text{の係数}]$$

$$b_{41} = p_{31} + c_1 \cdot q_0 \quad [x^1 \text{の係数}]$$

$$b_{40} = p_0 + c_0 \cdot q_0 \quad [x^0 \text{の係数}]$$

セレクタ1 (3) は、これらの計算結果をUregに格納する (図18: Uregのみ図示)。

【0059】

このようにして、 $b_4(x)$ が求められる。なお、 $b_4(x) = b_{43}x^3 + b_{42}x^2 + b_{41}x + b_{40}$ と記載するものとする。最後に、セレクタ1 (3) はUregの内容をYregとZregとに格納する (図19: YregとZregのみ図示)。

【0060】

次に、 $a_5(x) = Q(x^7 + b_4^2, a_4)$ を計算する。 a_4 は4次式であるから、 Q の計算においては $x^7 + b_4^2$ の3次以下の項は不要となる。 $b_4^2 = b_{43}^2x^6 + b_{42}^2x^4 + b_{41}^2x^2 + b_{40}^2$ であるから、 $b_{43}^2x^6 + b_{42}^2x^4 + x^7$ のみが使用されることになる。すなわち、以下の計算を実行するため、セレクタ2 (9) は、レジスタ群1から必要な値を取り出し、乗算器及び二乗器5に入力する。

$$(1) \quad p_{17} = 1 \quad [x^7 \text{の係数}]$$

$$p_{16} = b_{43}^2 \quad [x^6 \text{の係数}]$$

$$p_{15} = 0 \quad [x^5 \text{の係数}]$$

$$p_{14} = b_{42}^2 \quad [x^4 \text{の係数}]$$

セレクタ1 (3) は、これらの計算結果をUregに格納する (図20: Uregのみ図示)。

【0061】

次に、 a_4 による剩余計算を行う。より具体的には、以下の計算を実行する。

よって、セレクタ2(9)は、レジスタ群1から必要な値を取り出し、乗算器及び二乗器5に入力する。

(2-1)

$$\begin{aligned} p_{26} &= p_{16} + p_{17} \cdot a_{43} \\ &= p_{16} + a_{43} [x^6 \text{の係数}] \end{aligned}$$

$$\begin{aligned} p_{25} &= p_{15} + p_{17} \cdot a_{42} \\ &= a_{42} [x^5 \text{の係数}] \end{aligned}$$

$$\begin{aligned} p_{24} &= p_{14} + p_{17} \cdot a_{41} \\ &= p_{14} + a_{41} [x^4 \text{の係数}] \end{aligned}$$

なお、 $a_{53} = p_{17} [a_5 \text{の3次の項の係数}]$ となる。

セレクタ1(3)は、 $p_{17}=1$ を取り出し、これらの計算結果と共にUregに格納する(図21:Uregのみ図示)。なお、 $a_5(x) = a_{53}x^3 + a_{52}x^2 + a_{51}x + a_{50}$ とする。

【0062】

さらに、 a_4 による剩余計算を行う。より具体的には、以下の計算を実行する。よって、セレクタ2(9)は、レジスタ群1から必要な値を取り出し、乗算器及び二乗器5に入力する。

(2-2)

$$p_{35} = p_{25} + p_{26} \cdot a_{43} [x^5 \text{の係数}]$$

$$p_{34} = p_{24} + p_{26} \cdot a_{42} [x^4 \text{の係数}]$$

なお、 $a_{52} = p_{26} [a_5 \text{の2次の項の係数}]$ となる。

セレクタ1(3)は、 p_{17} 及び p_{26} を取り出し、これらの計算結果と共にUregに格納する(図22:Uregのみ図示)。

【0063】

さらに、 a_4 による剩余計算を行う。より具体的には、以下の計算を実行する

。よって、セレクタ2(9)は、レジスタ群1から必要な値を取り出し、乗算器及び二乗器5に入力する。

(2-3)

$$a_{50} = p_{34} + p_{35} \cdot a_{43} \quad [a_5 \text{の定数項}]$$

なお、 $a_{51} = p_{35}$ [a₅の1次の項の係数]となる。

セレクタ1(3)は、p₁₇及びp₂₆及びp₃₅を取り出し、これらの計算結果と共にUregに格納する(図23:Uregのみ図示)。これにてa₅が計算できた。

【0064】

(3)の処理では、セレクタ1(3)が、Uregに格納されたa₅(x)をXregに格納する(図24:Xregのみ図示)。

【0065】

次に、 $b_5(x) = (b_4 + 1) \bmod a_5(x)$ を計算する。b₄はYregとZregに格納されている。最初に(4)の処理として、セレクタ1(3)がb₄₃, b₄₂, b₄₁及びb₄₀+1をUregに格納する(図25:Uregのみ図示)。

【0066】

次に、a₅による剰余計算を行う。以下に必要な計算を詳細に記述する。よって、セレクタ2(9)は、必要な値をレジスタ群1から取り出し、乗算器及び二乗器5に入力する。

$$(5) \quad b_{52} = b_{42} + b_{43} \cdot a_{52} \quad [b_5 \text{の2次の項の係数}]$$

$$b_{51} = b_{41} + b_{43} \cdot a_{51} \quad [b_5 \text{の1次の項の係数}]$$

$$b_{50} = b_{40} + b_{43} \cdot a_{50} \quad [b_5 \text{の0次の項の係数}]$$

$$b_5(x) = b_{52}x^2 + b_{51}x + b_{50} \text{と表すものとする。}$$

セレクタ1(3)は、これらの計算結果をZregに格納する(図26:Zregのみ図示)。よって、Xreg及びZregにa₅とb₅が格納されることになる。なお、解としてa' = a₅, b' = b₅である。

【0067】

では、本発明のアルゴリズム（2倍算）を実施する場合の図1の回路の動作を説明する。図2の初期状態は、2倍算の場合あまり変わらない。但し、レジスタ a_2 及び b_2 は空となる。

【0068】

最初に $q(x) = Q(b_3, a_1)$ を計算するため、 $b_3(x) = b_1^2 + x \cdot (a_1 - x^3)^2$ を計算する。但し、 a_1 は3次式であるから、 $b_3(x)$ の3次以上の項を計算すればよい。より詳細に記述すると以下のようになる。セレクタ2(9)は、必要な値をレジスタ群1から取り出し、乗算器及び二乗器5に入力する。

$$(1) \quad b_{35} = c_2^2 \quad [x^5 \text{の係数}]$$

$$b_{34} = d_2^2 \quad [x^4 \text{の係数}]$$

$$b_{33} = c_1^2 \quad [x^3 \text{の係数}]$$

これらの計算結果をセレクタ1(3)はUregに格納する（図27：Uregのみ図示）。

【0069】

なお、

$$\begin{aligned} b_1^2 + x \cdot (a_1 - x^3)^2 &= \\ c_2^2 x^5 + d_2^2 x^4 + c_1^2 x^3 + d_1^2 x^2 + c_0^2 x + d_0^2 &= \\ b_{35} x^5 + b_{34} x^4 + b_{33} x^3 + b_{32} x^2 + b_{31} x + b_{30} \end{aligned}$$

である。

【0070】

次に、 $Q(b_3, a_1)$ を計算する。より詳細に記述すると以下のようになる。セレクタ2(9)は、必要な値をレジスタ群1から取り出し、乗算器及び二乗器5に入力する。

(2-1)

$$p_{14} = b_{34} + b_{35} \cdot c_2 \quad [x^4 \text{の係数}]$$

$$p_{13} = b_{33} + b_{35} \cdot c_1 \quad [x^3 \text{の係数}]$$

$$p_{12} = b_{35} \cdot c_0 \quad [x^2 \text{の係数}]$$

なお、 $a_2 = b_{35}$ である。 $q(x) = a_2 x^2 + a_1 x + a_0$ と記載する。

セレクタ1(3)は、 b_{35} をレジスタ群1から取り出し、これらの計算結果と共に、Uregに格納する(図28:Uregのみ図示)。

【0071】

同様に、 a_1 による剩余計算を行う。より詳細に記述すると以下のようになる。セレクタ2(9)は、必要な値をレジスタ群1から取り出し、乗算器及び二乗器5に入力する。

(2-2)

$$p_{23} = p_{13} + p_{14} \cdot c_2 \quad [x^3 \text{の係数}]$$

$$(p_{22} = p_{12} + p_{14} \cdot c_1 \quad [x^2 \text{の係数}])$$

なお、 $a_1 = p_{14}$ である。また、 $a_0 = p_{23}$ である。

セレクタ1(3)は、 a_2 及び p_{14} をレジスタ群1から取り出し、これらの計算結果と共に、Uregに格納する(図29:Uregのみ図示)。

【0072】

なお、同時に c_2^2 の逆数を求めるため、セレクタ2(9)は c_2^2 を乗算器及び二乗器5から受け取り、インバータ7に入力する。ここで、 $q^{-1} = 1 / c_2^2$ とする。

【0073】

$a_4(x) = q^2(x) + x$ をモニック化する必要があるので、以下の計算を実行する。セレクタ2(9)は、必要な値をレジスタ群1及びインバータ7から取り出し、乗算器及び二乗器5に入力する。

$$(3) \quad a_{43} = 0 \quad [x^3 \text{の係数}]$$

$$a_{42} = q_1^2 \cdot q^{-2} \quad [x^2 \text{の係数}]$$

$$a_{41} = 1 \cdot q^{-2} \quad [x \text{の係数}]$$

$$a_{40} = q_0^2 \cdot q^{-2} \quad [x^0 \text{の係数}]$$

これらの計算結果をセレクタ1(3)はXregに格納する(図30:Xregのみ図示)。なお、 $a_{44} = 1 \quad [x^4 \text{の係数}]$ であるからあえて記憶しておく必要はない。

い。 ($a_4(x) = x^4 + a_{43}x^3 + a_{42}x^2 + a_{41}x + a_{40}$) と記載する。

【0074】

次に、 $b_4 = (b_3 + 1) \bmod a_4$ を計算する。 a_4 は 4 次式であるから $(b_3 + 1)$ の 3 次以下の項は剩余計算の後に加算しても計算結果は変わらない。 Ureg の記憶位置の数及び乗算器の数の制限を考慮して以下のような計算を行う。なお、セレクタ 2 (9) は必要な値をレジスタ群 1 から取り出し、乗算器及び二乗器 5 に入力する。

$$(4) \quad b_{35} = c_2^2 \quad [x^5 \text{の係数}]$$

$$b_{34} = d_2^2 \quad [x^4 \text{の係数}]$$

$$b_{33} = c_1^2 \quad [x^3 \text{の係数}]$$

$$b_{32} = d_1^2 \quad [x^2 \text{の係数}]$$

これらの計算結果をセレクタ 1 (3) は Ureg に格納する (図 3-1 : Ureg のみ 図示)。

【0075】

そして a_4 による剩余計算を行う。但し、 b_3 の 1 次の項を加算する。より詳細に記述すれば以下のようない算計算を行う。セレクタ 2 (9) は必要な値をレジスタ群 1 から取り出し、乗算器及び二乗器 5 に入力する。

$$(5) \quad p_{14} = b_{34} + b_{35} \cdot a_{43} \\ = b_{34} \quad [x^4 \text{の係数}]$$

$$p_{13} = b_{33} + b_{35} \cdot a_{42} \quad [x^3 \text{の係数}]$$

$$p_{12} = b_{32} + b_{35} \cdot a_{41} \quad [x^2 \text{の係数}]$$

$$p_{11} = b_{35} \cdot a_{40} + c_0^2 \quad [x^1 \text{の係数}]$$

これらの計算結果をセレクタ 1 (3) は Ureg に格納する (図 3-2 : Ureg のみ 図示)。

【0076】

さらに a_4 による剩余計算を行う。但し、 b_3 の 0 次の項及び 1 を加算する。より詳細に記述すれば以下のようない算計算を行う。セレクタ 2 (9) は必要な値をレ

ジスタ群1から取り出し、乗算器及び二乗器5に入力する。

$$(6) \quad p_{23} = p_{13} + p_{14} \cdot a_{43} \quad [x^3 \text{の係数}]$$

$$p_{22} = p_{12} + p_{14} \cdot a_{42} \quad [x^2 \text{の係数}]$$

$$p_{21} = p_{11} + p_{14} \cdot a_{41} \quad [x^1 \text{の係数}]$$

$$p_{20} = p_{14} \cdot a_{40} + d_0^2 + 1 \quad [x^0 \text{の係数}]$$

これらの計算結果をセレクタ1(3)はYregとZregに格納する(図33:Yreg及びZregのみ図示)。これにて、 $b_4(x)$ が求まった。

【0077】

これ以降の計算は、通常の加算の場合と同じである。

【0078】

以上の処理を処理フローとして示すと図34のようになる。最初に、 $a_1(x)$, $b_1(x)$, $a_2(x)$, b_2 が入力される(ステップ100)。2倍算の場合には、 $a_1(x)$ 及び $b_1(x)$ のみ入力される。次に、通常の加算か2倍算かで処理を切り換える(ステップ110)。もし、2倍算の場合には、 $q^- = 1 / c_2^2$ を計算する(ステップ120)。また、 $q(x) = Q(b_3, a_1)$ をUregに格納する(ステップ130)。図1のような回路ではステップ120及び130は同時に実行される。そして、モニック化された $a_4(x) = q^{-2} (Ureg^2 + x)$ を計算して、Xregに格納する(ステップ140)。一方、ステップ110で通常の加算であると判断された場合には、 a_1 及び a_2 の最大公約多項式を計算する。最大公約多項式が1でない場合は、本発明では取り扱わない。そして、 $s_1(x)a_1(x) + s_2(x)a_2(x) = 1$ である s_1 を計算し、Yregに格納する(ステップ150)。次に $q(x) = s_1(b_1 + b_2) \bmod a_2$ を計算し、Zregに格納する(ステップ160)。そして、 $q^- = 1 / q_2^2$ を計算する(ステップ170)。また、 $Q(q^2 a_1, a_2) + x + c_2 + e_2$ を計算し、Uregに格納する(ステップ180)。ステップ170及びステップ180は図1の回路では同時に実行する。そして、モニック化された $a_4(x) = q^- Ureg$ を計算し、Xregに格納する(ステップ190)。

【0079】

以下の処理は、通常の加算及び2倍算と共通である。 $b_4(x) = (b_3 + 1) \bmod$

$d a_4$ を計算し、Yreg及びZregに格納する（ステップ200）。但し、通常の加算の場合と2倍算の場合では、 b_3 の定義が異なる。そして、 $a_5(x) = Q(x^7 + b_2^2, a_4)$ を計算し、Xregに格納する（ステップ210）。最後に、 $b_5(x) = (b_4 + 1) \bmod a_5$ を計算し、Zregに格納する（ステップ220）。

【0080】

図3.4のような処理を通常のコンピュータ（例えば図3.5）のためのコンピュータ・プログラムにて実装することができる。但し、二乗算は通常のコンピュータでは高速に実行できないので、処理速度向上には限界がある。

【0081】

なお、このような本発明のアルゴリズムを実行するような装置及びプログラムを実装して暗号化装置又は復号化装置又はそれらを含む暗号システムを構築することができる。

【0082】

【効果】

ヤコビ多様体の群演算を少ない演算量にて実現することができた。

【0083】

また、ヤコビ多様体の群演算を少ないハードウェア量にて実施可能とすることもできた。

【0084】

【計算量の評価】

本発明のアルゴリズム（通常の加算及び2倍算）の乗算の実行回数について評価する。以下mは乗算器1つが乗算1回を行うことを、Mは複数の乗算器が同時に乗算を1回実行することを表すと定義する。すなわち、mは乗算回数を表すために用いられ、Mは乗算器群が呼び出された回数を表す。またIは逆元を求める計算1回を表す。以後I, M, mを用いて計算量を表すものとする。例えば、I + 2mは1回の逆元演算と2回の乗算が行われることを表す。次の表1及び表2は加算と2倍算の計算量をまとめたものである。

【表1】

計算	計算量	呼出回数	時間
GCD	$3I + 23m$	$3I + 9M$	$3t(I) + 9t(M)$
$q(x)$	$15m$	$4M$	$4t(M)$
$a_4(x)$	$I + 20m$	$I + 6M$	$t(I) + t(M)$
$b_4(x)$	$17m$	$5M$	$5t(M)$
$a_5(x), b_5(x)$	$6m$	$3M$	$3t(M)$
合計	$4I + 81m$	$4I + 27M$	$4t(I) + 22t(M)$

【表2】

計算	計算量	呼出回数	時間
$q(x)$	$3m$	$2M$	0
$a_4(x)$	$I + 2m$	$I + M$	$t(I) + t(M)$
$b_4(x)$	$8m$	$2M$	$2t(M)$
$a_5(x), b_5(x)$	$6m$	$3M$	$3t(M)$
合計	$I + 19m$	$I + 8M$	$t(I) + 6t(M)$

なお表1及び表2の中では $t(I)$ は逆元を計算するための時間を、 $t(M)$ は乗算を計算する時間を表す。また 2^n 乗算は1クロック・サイクルで実行できるものとして無視している。

【0085】

表1(加算)では $t(I) > 5t(M)$ であるものと仮定している。これにより $a_4(x)$ の計算中に $a_4(x)$ をモニック化するための逆元の計算を同時にを行うことがで

きる。さらに表2（2倍算）では $t(I) > 2t(M)$ であるものと仮定している。これにより $q(x)$ の計算と $a_4(x)$ をモニック化するために必要な逆元の計算を並列に行うことができる。

【0086】

表1及び表2に対し、T. Itoh, S. Tsujii, "A Fast Algorithm for Computing Multiplicative Inverse in GF(2^m) Using Normal Bases," Inform. and Comput., vol.83, No. 1, pp.171-177, (1989)記載の方法（以下、伊東一辻井方法と呼ぶ）で GF(2⁵⁹) 上で $t(I)=8t(M)$ が成り立つことを用いると、通常の加算の場合、計算量は 113m であり、時間は 54t(M) である。また、2倍算の場合、計算量は 27m であり、時間は 14t(M) となる。一方、酒井康行、石塚裕一、櫻井幸一、"安全な超楕円曲線暗号の構成とその実装," SCIS'98-10.1.B, Jan. 1998 (以下、文献1と呼ぶ) で得られた結果は表3のとおりである。

【表3】

	加 算		2倍算	
	乗算	逆元演算	乗算	逆元演算
$g = 0$	3	1	3	1
$g = 3$	401	0	265	0
$g = 11$	17477	0	10437	0

【0087】

表3と表1及び表2を比べると、通常の加算の場合、計算量で 3.5 倍、時間で 7 倍、2倍算の場合、計算量で 10 倍、時間で 19 倍、本発明のアルゴリズムの方が良い。また 7 個の乗算器の使用効率は、通常の加算で 0.572、2倍算で 0.45 である。よって本発明のアルゴリズムは従来の手法と比べて効率的に計算が実行されると同時に高い並列度を持っている。

【0088】

[処理性能の評価]

表1及び表2に基づいて160bit程度の整数倍を行うのに必要な時間を計算すると表4となる。なお、2倍算は160回、加算は80回行うものと仮定している。

【表4】

動作周波数	1回の乗算に必要なクロック		
	ケースA $t(M)=59$ クロック	ケースB $t(M)=8$ クロック	ケースC $t(M)=1$ クロック
20MHz	19.35ms	2.624ms	0.328ms
40MHz	9.68ms	1.312ms	0.164ms
80MHz	4.84ms	0.656ms	0.082ms

【0089】

一方文献1のソフトウェアによる実装では、Alpha 21164 (250MHz) (AlphaはDigital Equipment Corp.の商標) を用い、加算には $500\mu s$ 、2倍算には $50\mu s$ 、160bitの整数倍には118msの処理時間を要した。この結果と比べると本発明のアルゴリズムをハードウェア実装したものは、動作周波数20MHz、Case Aの場合で約5倍、Case Bの場合で約50倍、Case Cの場合で約360倍高速に処理を行う。RSA暗号において専用ハードウェアによる計算と動作周波数が10倍程度異なる汎用のMPUによる計算の処理時間の比が5倍程度であることを考えると、超楕円曲線暗号と本発明のアルゴリズムは実用上ハードウェア実装にかなり適していると言える。

【0090】

また安全性で同等と考えられる160bit鍵楕円曲線暗号では、技術速報、NIKKEI ELECTRONICS, 1998.3.23, (no.712) pp. 23によると20MHzの動作周波数で署名を行うのに最大3.6msの時間が、また鳥居直哉、岡田壮一、長谷部高行、"楕円曲線暗号チップの試作," 信学ソサイエティ大, A-7-1, Oct. 1998

によると27Kゲートのハードウェアを20MHzの動作周波数で、60msの平均処理時間がかかることが報告されている。それらと比べると提案アルゴリズムは同等か、数倍高速に処理を行う。

【0091】

ここで処理性能と消費電力について楕円曲線暗号 ($g = 1$) と超楕円曲線暗号 ($g > 1$ となる任意の g) を比較する。超楕円曲線暗号の演算は楕円曲線暗号と比べて複雑である。しかし定義体はおよそ $1/g$ のものを用いることができる。一般に定義体を $G F (2^n)$ とすると乗算器のハードウェア量は n の2乗に、消費電力も n の2乗に比例し、演算のスピードは $1/(1 - (\log_n g))$ に比例すること。よって、乗算器性能の種数に対する依存性は $g^4 (1 + \log_n g + (\log_n g)^2 + \dots)$ となる。それに対し演算量の増加は g^2 に比例する。よって漸近的には超楕円曲線暗号のほうが $g^2 (1 + \log_n g + \dots)$ だけ有利である。またハードウェア実装の点から見ても超楕円曲線暗号の方が g 倍の並列性を実現できることも利点である。

【0092】

[ゲートアレーにマッピングした場合の評価]

上の説明では評価を行うために $t(M)$ と動作周波数を用いた。最大動作周波数を求めるためには具体的に回路設計を行い、半導体のテクノロジーにマッピングを行わなければならない。そこで8クロック (clock) で乗算器が計算を行う表4のCase Bの場合についてVHDL (IEEE std 1076-1987) を用いて設計し、有効チャネル長 $L_{eff} = 0.27 \mu m$ のCMOSゲートアレー・テクノロジー (IBM CMOS 5SE) にマッピングした場合の評価を行った。その結果、レジスタ間最大遅延12ns (最大動作周波数83MHzに対応する)、ハードウェアサイズ約140Kセル (cell) の結果を得た。個々のブロックのサイズについては表5に示す。

【表5】

図1のブロック	サイズ(セル)	
乗算器	34265セル	乗算器7個
二乗器	1344セル	二乗器3個
インバータ	27414セル	
レジスタ群	18408セル	
コントローラ	9749セル	59ビットレジスタ26個 (係数12個を含む)
セレクタ1	37140セル	
セレクタ2	17402セル	
合計	145722セル	

なお、全体のセル数が140Kセルとなったのは、個々のブロックを結合した後で回路全体のタイミングを最適化することにより約5Kセル減らすことにより実現されたものである。これらの動作周波数・サイズはRSA暗号などの暗号化VLSIなどと比べても十分実用的な数字である。なおGF(2^{59})の原始多項式として $p(x) = x^{59} + x^6 + x^5 + x^4 + x^3 + x + 1$ を用いた。理由はGF(2^{59})には、オプティマル・ノーマル・ベース(Optimal Normal Bases: GF(2^n)の正規基底の中で乗算結果1bitが $2^n - 1$ 個の項の和であらわせるもの)が存在しないこと、また基礎体がGF(2)の時、円分体は偶数次の拡大体にしか存在しないからである。

【図面の簡単な説明】

【図1】

本発明の全体のブロック図である。

【図2】

本発明のアルゴリズム(通常の加算)を実施する際の、レジスタ群1の初期状態を示す図である。

【図3】

$q(x) = s_1(b_1 + b_2) \bmod a_2$ の途中結果を格納している Ureg の状態を示す図である。

【図4】

$q(x) = s_1(b_1 + b_2) \bmod a_2$ の途中結果を格納している Ureg の状態を示す図である。

【図5】

$q(x) = s_1(b_1 + b_2) \bmod a_2$ の途中結果を格納している Ureg の状態を示す図である。

【図6】

$q(x)$ の最終結果を格納している Zreg の状態を示す図である。

【図7】

$a_4(x) = Q(q^2 a_1, a_2) + x + c_2 + e_2$ の途中結果を格納している Ureg の状態を示す図である。

【図8】

$a_4(x) = Q(q^2 a_1, a_2) + x + c_2 + e_2$ の途中結果を格納している Ureg の状態を示す図である。

【図9】

$a_4(x) = Q(q^2 a_1, a_2) + x + c_2 + e_2$ の途中結果を格納している Ureg の状態を示す図である。

【図10】

$a_4(x) = Q(q^2 a_1, a_2) + x + c_2 + e_2$ の途中結果を格納している Ureg の状態を示す図である。

【図11】

$a_4(x) = Q(q^2 a_1, a_2) + x + c_2 + e_2$ の途中結果を格納している Ureg の状態を示す図である。

【図12】

$a_4(x) = Q(q^2 a_1, a_2) + x + c_2 + e_2$ の途中結果を格納している Ureg の状態を示す図である。

【図13】

モニック化された $a_4(x)$ の最終結果を格納している Xreg の状態を示す図である。

【図 1 4】

$b_4(x) = (q \cdot a_1 + b_1 + 1) \bmod a_4$ の途中結果を格納している Ureg の状態を示す図である。

【図 1 5】

$b_4(x) = (q \cdot a_1 + b_1 + 1) \bmod a_4$ の途中結果を格納している Ureg の状態を示す図である。

【図 1 6】

$b_4(x) = (q \cdot a_1 + b_1 + 1) \bmod a_4$ の途中結果を格納している Ureg の状態を示す図である。

【図 1 7】

$b_4(x) = (q \cdot a_1 + b_1 + 1) \bmod a_4$ の途中結果を格納している Ureg の状態を示す図である。

【図 1 8】

$b_4(x) = (q \cdot a_1 + b_1 + 1) \bmod a_4$ の途中結果を格納している Ureg の状態を示す図である。

【図 1 9】

$b_4(x)$ の最終結果を格納している Yreg と Zreg の状態を示す図である。

【図 2 0】

$a_5(x) = Q(x^7 + b_4^2, a_4)$ の途中結果を格納している Ureg の状態を示す図である。

【図 2 1】

$a_5(x) = Q(x^7 + b_4^2, a_4)$ の途中結果を格納している Ureg の状態を示す図である。

【図 2 2】

$a_5(x) = Q(x^7 + b_4^2, a_4)$ の途中結果を格納している Ureg の状態を示す図である。

【図 2 3】

$a_5(x) = Q(x^7 + b_4^2, a_4)$ の途中結果を格納している Ureg の状態を示す図である。

【図 24】

$a_5(x)$ の最終結果を格納している Xreg の状態を示す図である。

【図 25】

$b_5(x) = (b_4 + 1) \bmod a_5(x)$ の途中結果を格納している Ureg の状態を示す図である。

【図 26】

$b_5(x)$ の最終結果を格納している Zreg の状態を示す図である。

【図 27】

$q(x) = Q(b_3, a_1)$ の途中結果を格納している Ureg の状態を示す図である。

【図 28】

$q(x) = Q(b_3, a_1)$ の途中結果を格納している Ureg の状態を示す図である。

【図 29】

$q(x)$ の最終結果を格納している Ureg の状態を示す図である。

【図 30】

モニック化された $a_4(x) = q^2(x) + x$ を格納している Xreg の状態を示す図である。

【図 31】

$b_4 = (b_3 + 1) \bmod a_4$ の途中結果を格納している Ureg の状態を示す図である。

【図 32】

$b_4 = (b_3 + 1) \bmod a_4$ の途中結果を格納している Ureg の状態を示す図である。

【図 33】

b_4 の最終結果を格納している Yreg と Zreg の状態を示す図である。

【図 34】

本発明のアルゴリズムのフローを示す図である。

【図35】

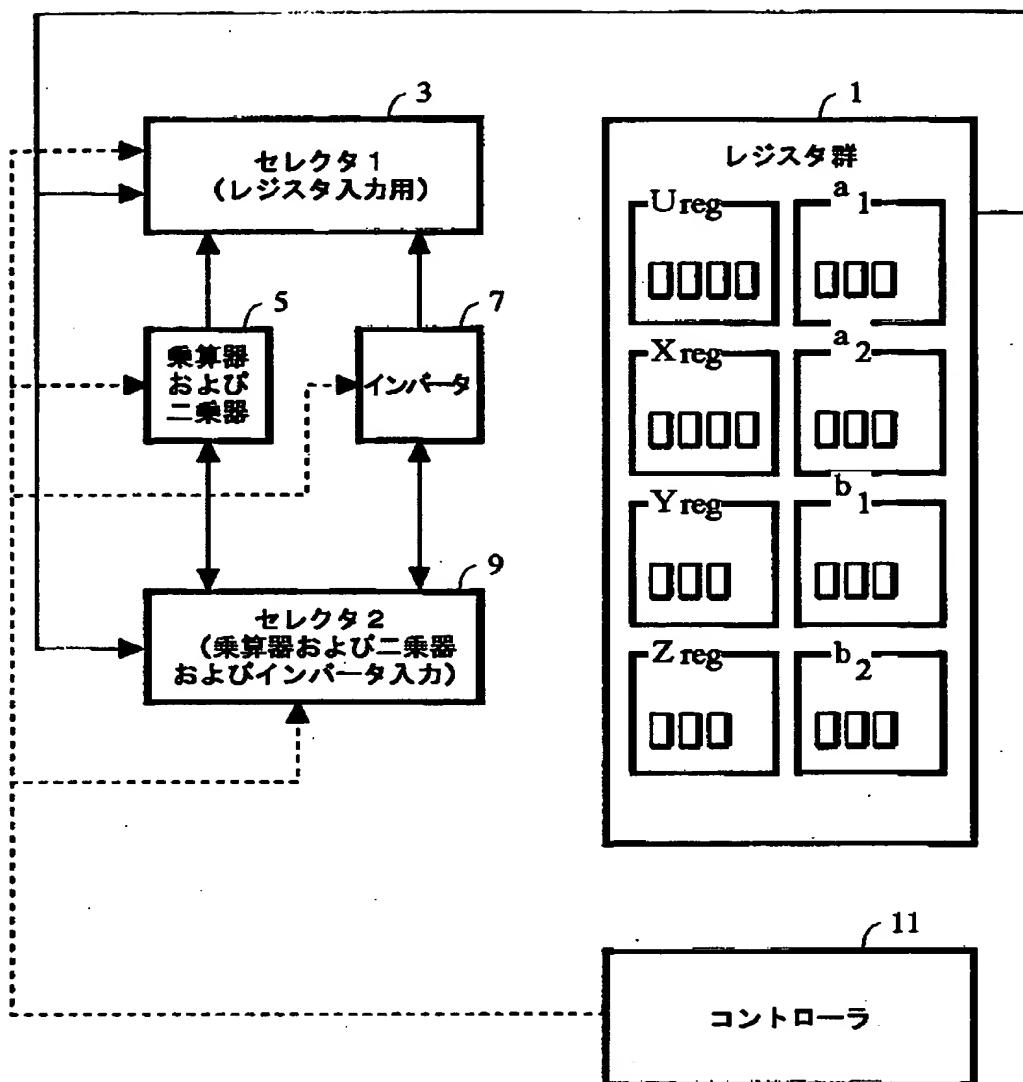
通常のコンピュータの構成例を示す図である。

【符号の説明】

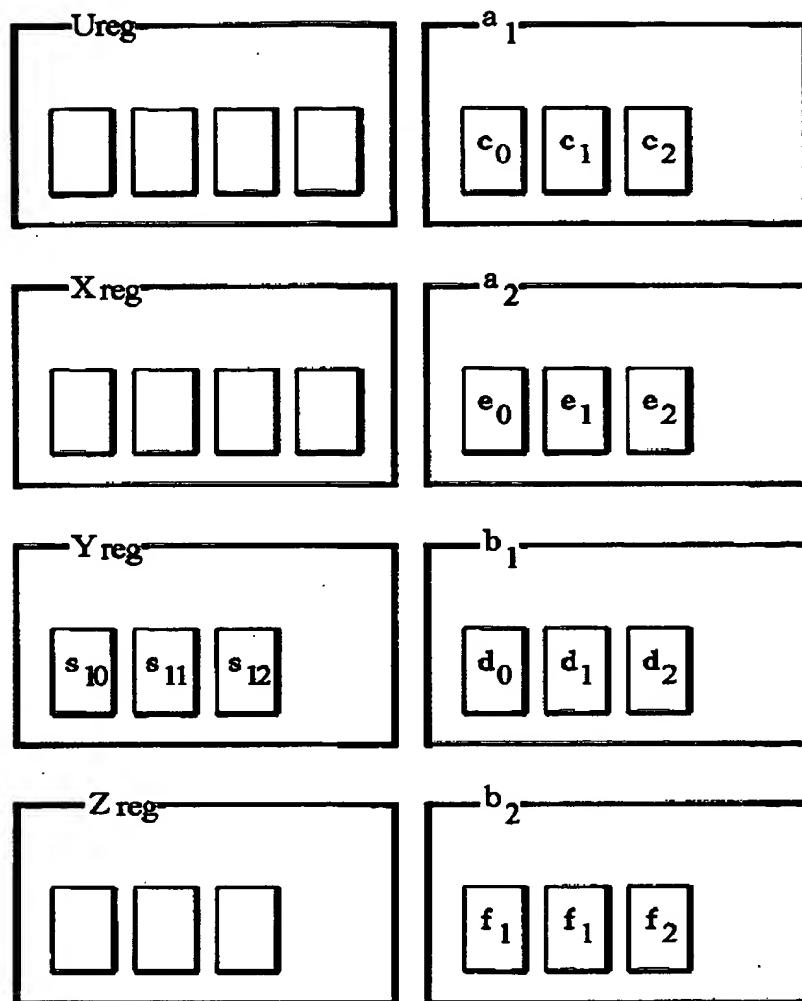
- 1 レジスタ群
- 3 セレクタ1
- 5 乗算器及び二乗器
- 7 インバータ
- 9 セレクタ2
- 11 コントローラ

【書類名】 図面

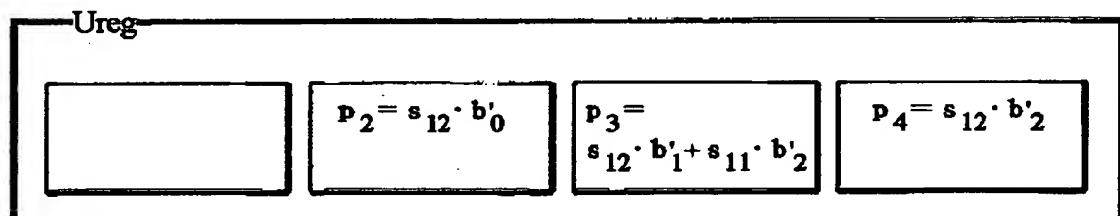
【図1】



【図2】



【図3】



【図4】

Ureg

$p'_0 = s_{10} \cdot b'_0$	$p'_1 = p_4 \cdot e_0$	$p'_2 =$ $p_2 + p_4 \cdot e_1$	$p'_3 =$ $p_3 + p_4 \cdot e_2$
----------------------------	------------------------	-----------------------------------	-----------------------------------

【図5】

Ureg

	$p''_0 =$ $p'_0 + p'_3 \cdot e_0$	$p''_1 =$ $p'_1 + p'_3 \cdot e_1$	$p''_2 =$ $p'_2 + p'_3 \cdot e_2$
--	--------------------------------------	--------------------------------------	--------------------------------------

【図6】

Zreg

$q_0 = p''_0$	$q_1 =$ $p''_1 + (s_{11} \cdot b'_0 + s_{10} \cdot b'_1)$	$q_2 =$ $p''_2 + (s_{11} \cdot b'_1 + s_{10} \cdot b'_2)$
---------------	--	--

【図7】

Ureg

$p_4 =$ $q_1^2 \cdot c_2 + q_2^2 \cdot c_0$	$p_5 =$ $q_2^2 \cdot c_1 + q_1^2$	$p_6 = q_2^2 \cdot c_2$	$p_7 = q_2^2$
--	--------------------------------------	-------------------------	---------------

【図 8】

Ureg

$p'_3 = q_1^2 \cdot c_1 + q_0^2$	$p'_4 = p_3 + p_7 \cdot e_0$	$p'_5 = p_4 + p_7 \cdot e_1$	$p'_6 = p_6 + p_7 \cdot e_2$
----------------------------------	------------------------------	------------------------------	------------------------------

【図 9】

Ureg

$a'_{43} = p'_6$	$p''_3 = p'_3 + p'_6 \cdot e_0$	$p''_4 = p'_4 + p'_6 \cdot e_1$	$p''_5 = p'_5 + p'_6 \cdot e_2$
------------------	---------------------------------	---------------------------------	---------------------------------

【図 10】

Ureg

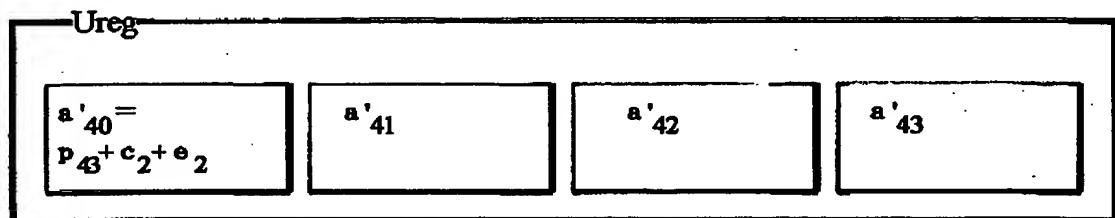
$a'_{42} = p''_5$	a'_{43}	$p_{33} = p''_3 + p''_5 \cdot e_1$	$p_{34} = p''_4 + p''_5 \cdot e_2$
-------------------	-----------	------------------------------------	------------------------------------

【図 11】

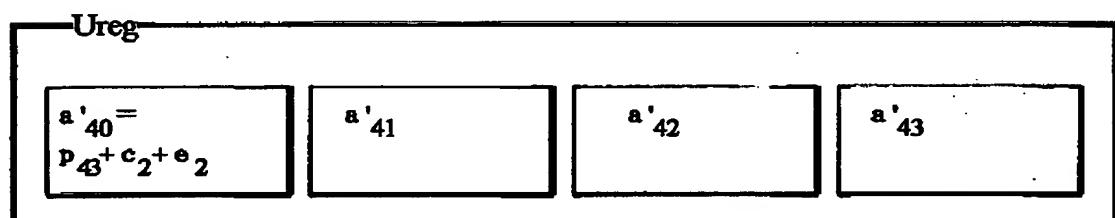
Ureg

$a'_{41} = p_{34} + 1$	a'_{42}	a'_{43}	$p_{43} = p_{33} + p_{34} \cdot e_2$
------------------------	-----------	-----------	--------------------------------------

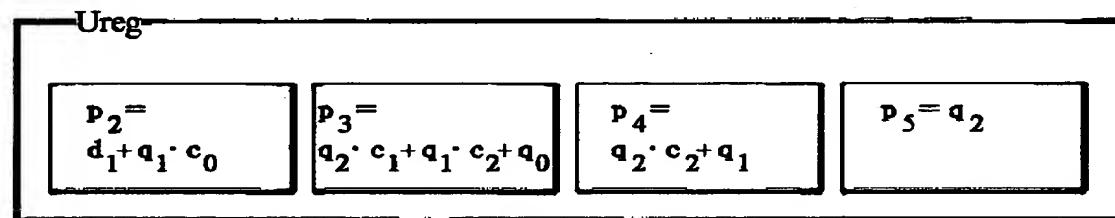
【図12】



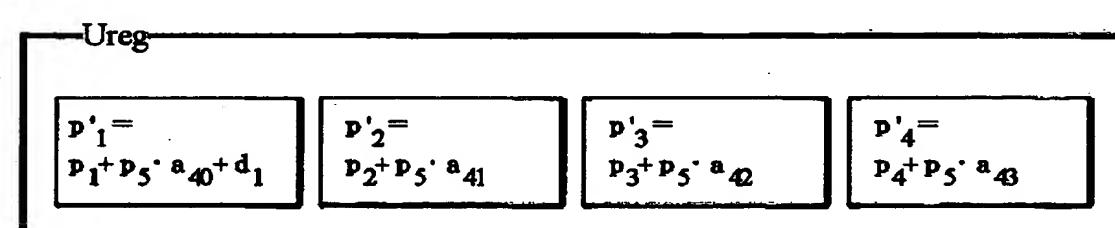
【図13】



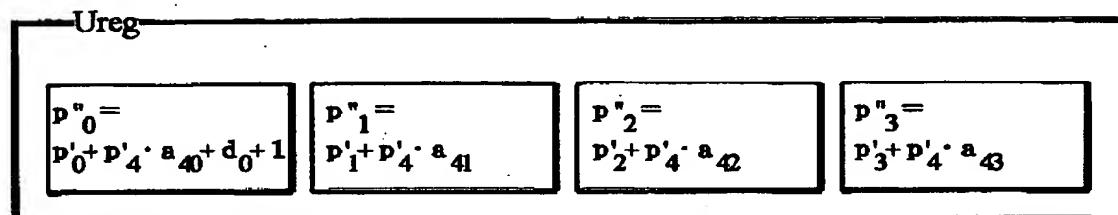
【図14】



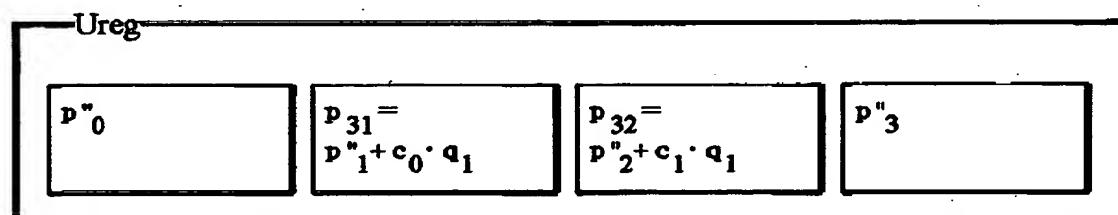
【図15】



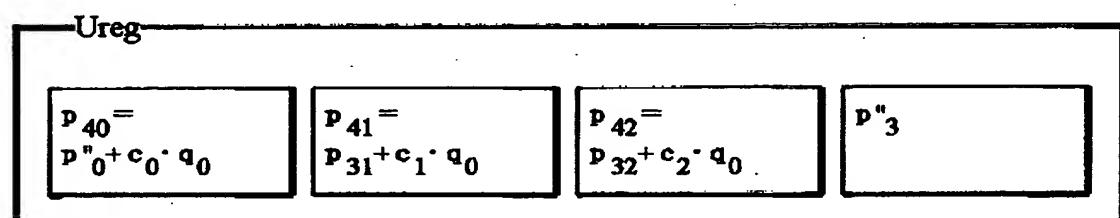
【図 1 6】



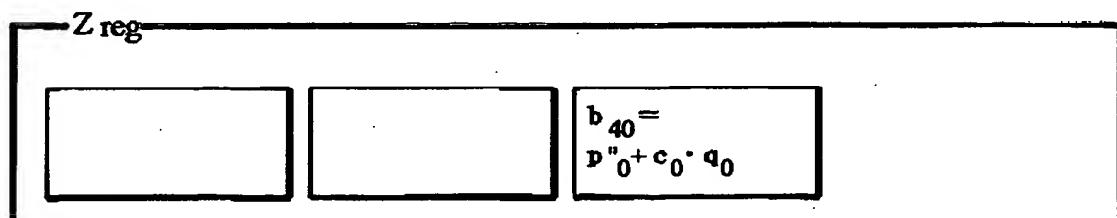
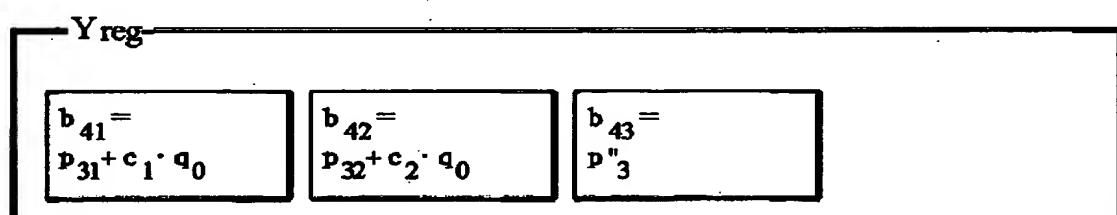
【図 1 7】



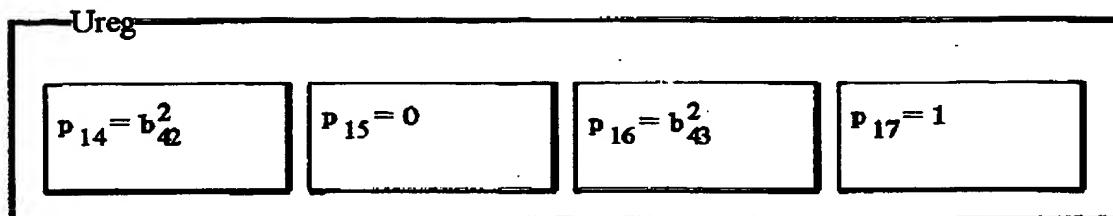
【図 1 8】



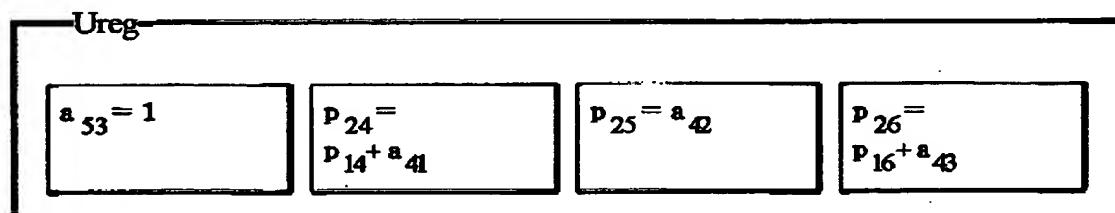
【図 1 9】



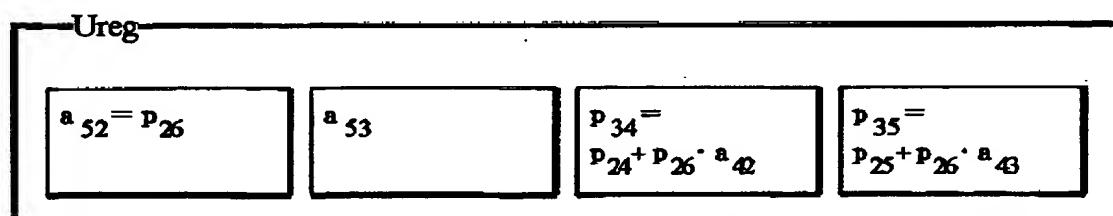
【図20】



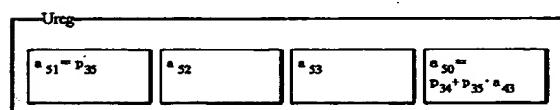
【図21】



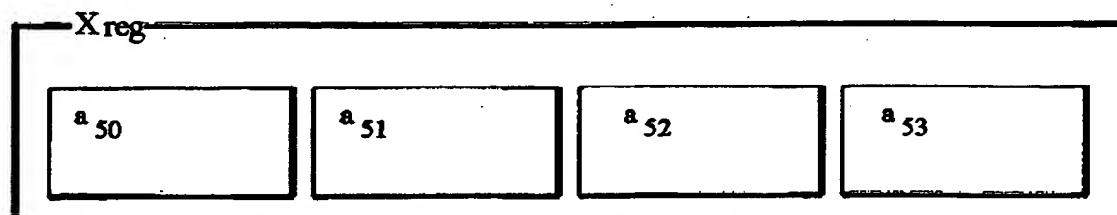
【図22】



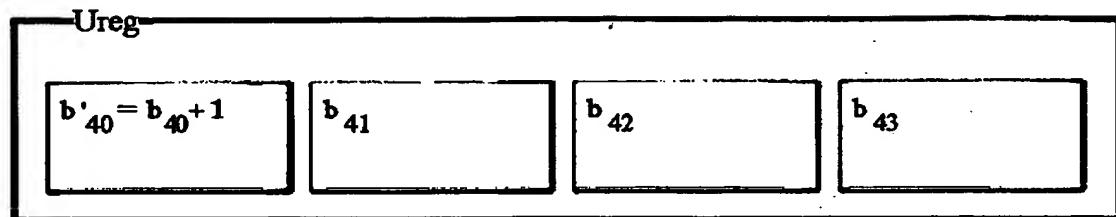
【図23】



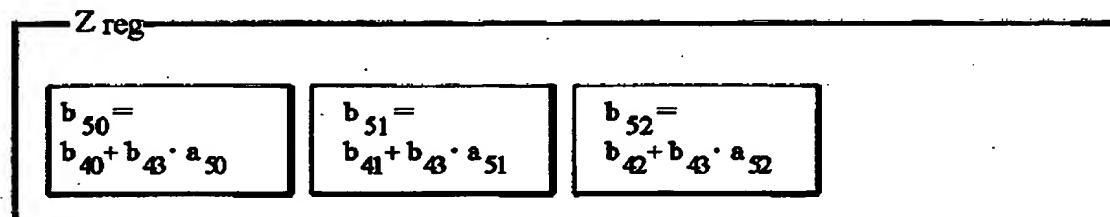
【図24】



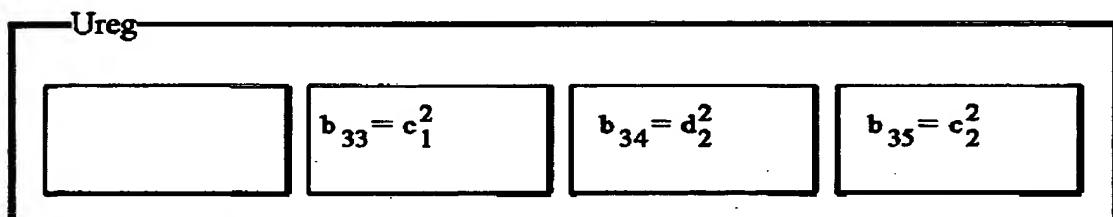
【図25】



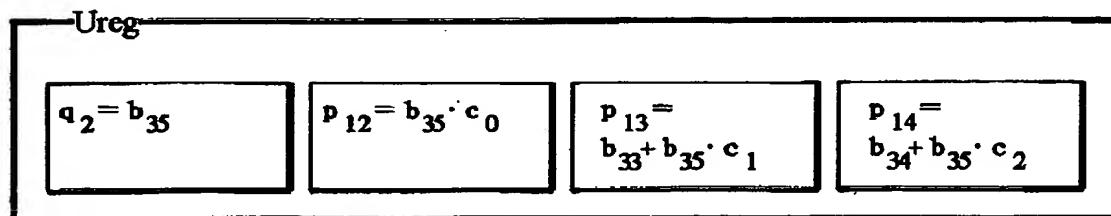
【図26】



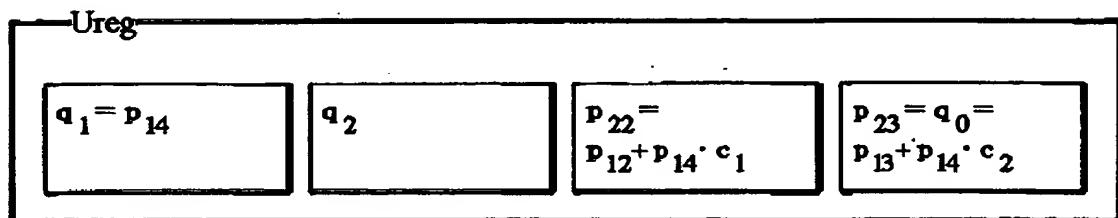
【図27】



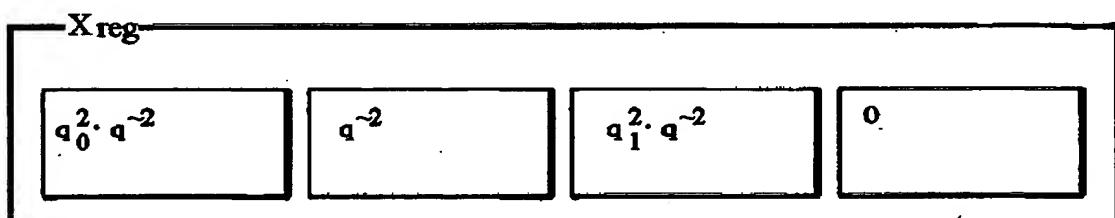
【図28】



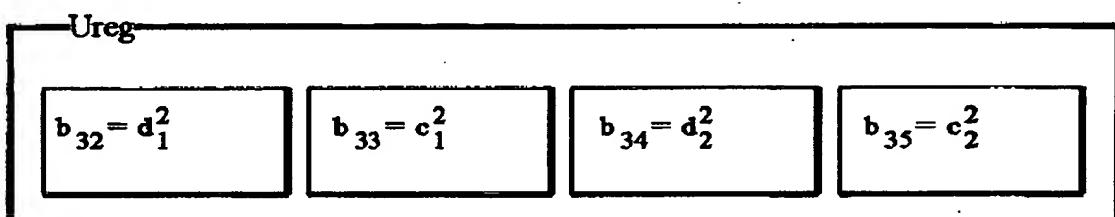
【図29】



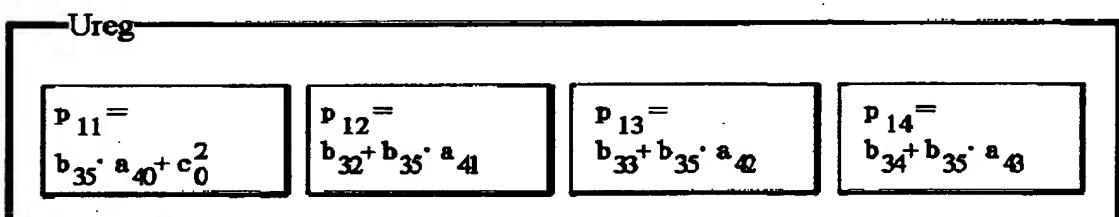
【図30】



【図31】



【図32】



【図33】

Y reg

$$p_{21} = \\ p_{11} + p_{14} \cdot a_{41}$$

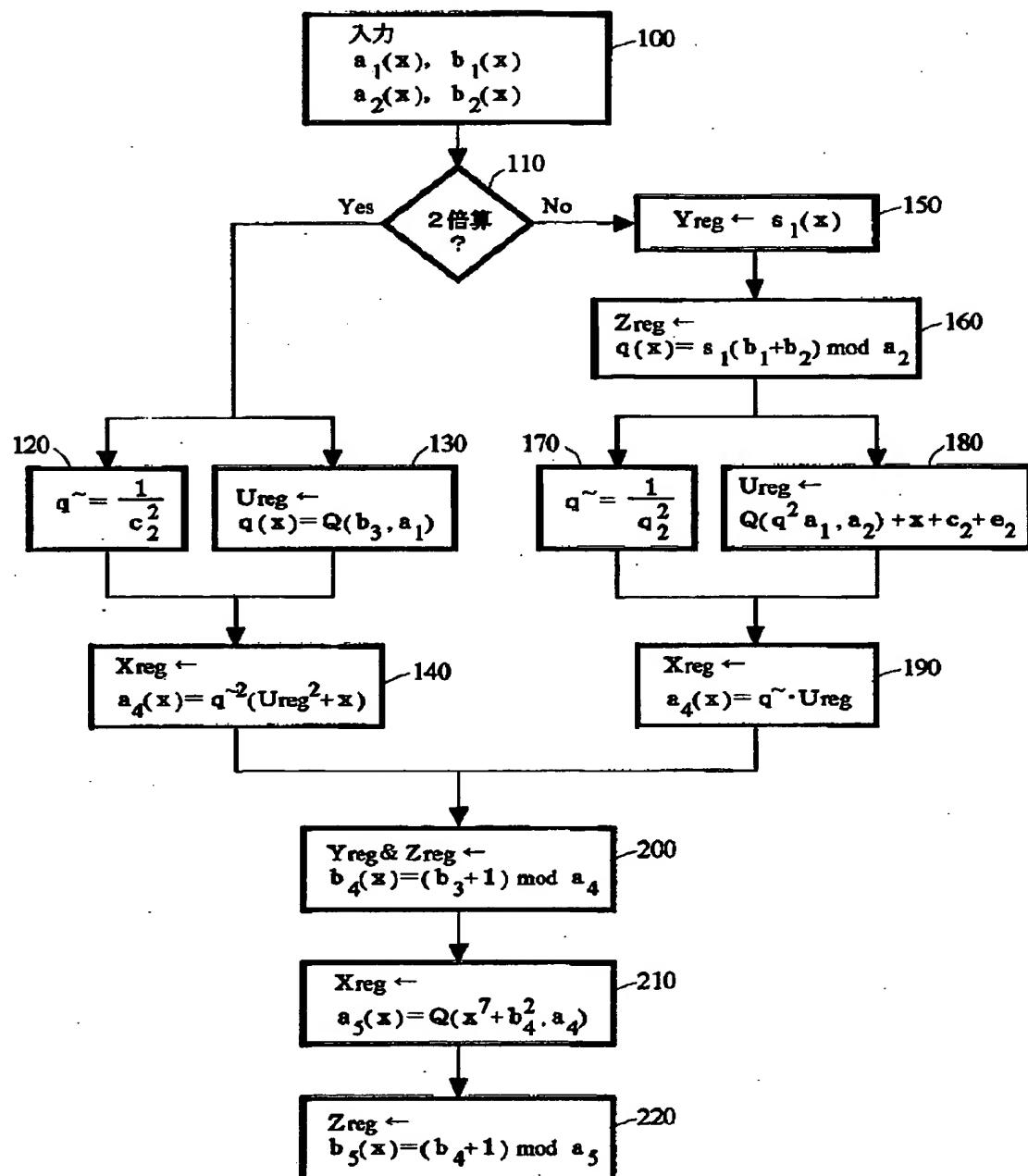
$$p_{22} = \\ p_{12} + p_{14} \cdot a_{42}$$

$$p_{23} = \\ p_{13} + p_{14} \cdot a_{43}$$

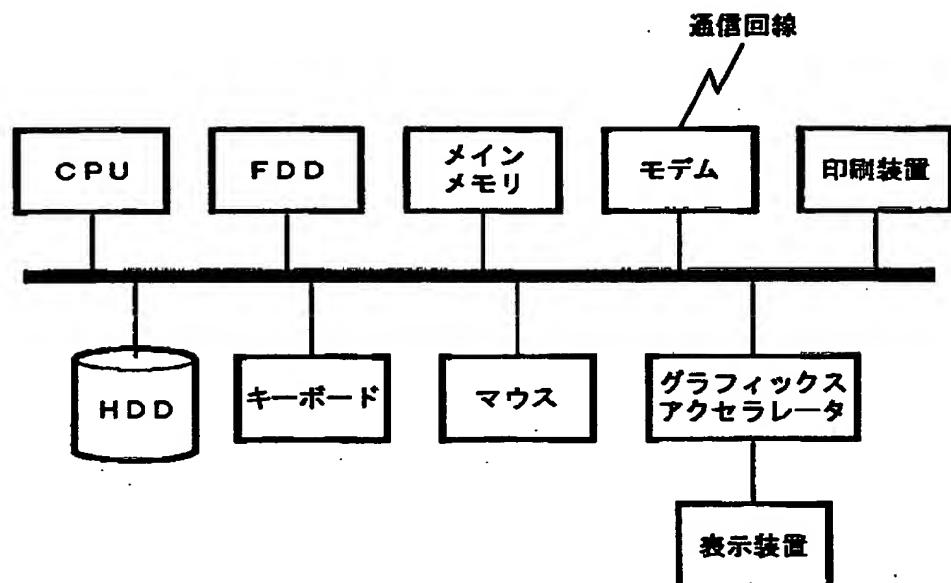
Z reg

$$p_{11} = \\ p_{14} \cdot a_{40} + d_0^2 + 1$$

【図34】



【図35】



【書類名】 要約書

【要約】

【課題】

ヤコビ多様体における群演算を少ない演算量にて実現すること。

【解決手段】

$G F (2^n)$ 上で定義される超楕円曲線 $y^2 + y = f(x)$ のヤコビ多様体の因子 $D_1 = g.c.d. (a_1(x), y - b_1(x))$ 及び $D_2 = g.c.d. (a_2(x), y - b_2(x))$ に対し群演算を実施する装置は、 $a_1(x)$ 、 $a_2(x)$ 、 $b_1(x)$ 及び $b_2(x)$ を格納する手段と、 $\text{GCD}(a_1(x), a_2(x)) = 1$ (GCD は最大公約多項式) である場合における $s_1(x) a_1(x) + s_2(x) a_2(x) = 1$ となる $s_1(x)$ を用いて、 $q(x) = \{s_1(b_1(x) + b_2(x))\} \bmod a_2(x)$ を計算する手段と有する。このように新たな関数 $q(x)$ を設けることにより、計算量が減少し、且つハードウェア量も少なくて済む。なお、 $D_1 = D_2$ の場合には、 $a_1(x)$ 及び $b_1(x)$ を格納する手段と、 $q(x) = Q(b_1^2(x) + f(x) \bmod a_1^2(x), a_1)$ ($Q(A, B)$ は A を B で割ったときの商) を計算する手段を設ける。

【選択図】 図 1

認定・付加情報

特許出願の番号 平成11年 特許願 第007384号
受付番号 59900029759
書類名 特許願
担当官 第七担当上席 0096
作成日 平成11年 1月24日

＜認定情報・付加情報＞

【提出日】 平成11年 1月14日

次頁無

出願人履歴情報

識別番号 [390009531]

1. 変更年月日 1990年10月24日

[変更理由] 新規登録

住 所 アメリカ合衆国10504、ニューヨーク州 アーモンク (番地なし)

氏 名 インターナショナル・ビジネス・マシーンズ・コーポレイション